



**SRI KRISHNA INSTITUTIONS
(AN AUTONOMOUS INSTITUTION)
(APPROVED BY AICTE AND AFFILIATED TO
ANNA UNIVERSITY CHENNAI)
ACCREDITED BY NAAC WITH "A" GRADE**



AUTOMOBILE SERVICING AND ASSISTS

AUTO-CARE

TEAM-40

SUBMITTED BY:

RAJESH KRISHNA R

NANDHA KUMAR R P

VIVIN VENKATESHWARAN S





JAYASHREE RAVICHANDRAN

DINESH KUMAR V

**DATE:
02-08-2020**

REVIEWED BY:

TABLE OF CONTENTS

1. Title	3
2. Abstract	3
3. Objective	3
4. Existing System	4
5. Proposed System	4
6. High Level Documentation	5
7. Low Level Documentation	10
 Control Flow Diagram	10
 Algorithm	11
 Pseudo code	20
 Data Flow Diagram	30
8. Source code	32
9. Output Screenshots	211
10. Test cases	224
11. Target	251

TITLE OF THE PROJECT

Auto care - An automobile servicing and assist system.

ABSTRACT

AUTO-CARE is a window based application which provides the user with a convenient way to service their vehicles. It also provides ecommerce solutions for vehicles. The purpose of our project is to reduce the difficulties while servicing the vehicle.

The problem faced by many of us during the time of servicing our vehicle is self transportation.

There are various solutions to resolve this problem such as providing the user with temporary vehicle, free pickup and drop facility. But we have several difficulties with these solutions as they are immoderate.

Our ultimate aim is to provide an efficient and authentic way by pre-booking the services of vehicles through our provided application.

And also we came up with the new idea of an immediate assist through which the user can straight away access the service centre with no time.

We make things reliable and best for the users through our application.

OBJECTIVES

- a) To promote a user-friendly application.
- b) To ensure the availability of the service time in prior by pre booking it in order to make ourselves opt to that particular time.
- c) To provide independent and separate domains for user and admin.
- d) To set simple and helpful access of the application in the desperate situation (i.e. breakdown other malfunctioning of vehicles etc.).
- e) To provide transparency in selling goods (vehicle parts) online.

EXISTING SYSTEM

In the existing system, the user has problems in self transportation (i.e.) the user has to leave his vehicle for service in the service station and go back through other means of transport.

This entire process kills the time in which the user could have done something productive. And maintaining data such as details of the dealers, purchases, sales details over a long period of time is quite difficult.

PROPOSED SYSTEM

In this proposed system, the user can ensure the availability of the service time of vehicle in advance by pre-booking. So that the user can adhere to the service timing.

In a critical situation like malfunctioning of the vehicle, we have provided immediate assist functionality at any time with the necessary prerequisites.

This application has an e-commerce mode for buying spare parts online if required for the users.

All the data will be stored in the database in which there will have not the scope of losing the stored information. Complete data storing and retrieving is achieved in this application which is 100% secure.

HIGH LEVEL DOCUMENTATION

1. TECHNICAL REQUIREMENTS

The automobile servicing and assists system is a window application which is developed in Java .It is implemented by using NetBeans and MySQL. This application uses JAVA AWT, Swings, SQL and Xampp.

2. LIST OF TABLES:

Users
Admin
Appointments
Ecommerce
Assist
Assist pin code
Invoice
Items
Schedule
State
District
Vehicle
Workers
Workshop

3. SIGN-UP/LOGIN:

This module is for the user as well as admin, admin's being individual workshops.

USER AND ADMIN REGISTRATION:

Both the User and Admin have to register before they go about their business; every detail collected is sent to the respective database tables. And the other modules use them as they require.

The only difference between user and admin registration is that the admin's are individual workshops or Auto Mobile service shops.

USER LOGIN:

It is mandatory that user should enter the username and password before accessing the site. Entered data is checked in the database.

If the user name matches the password then the entered data is correct. Then the user is allowed to access our booking site.

ADMIN LOGIN:

Admin is given an independent platform to monitor and authorize everything. Similar to the user login, admin is provided with the user name and password.

It is then checked in the database and assures that the entered username and password matches the database. Once they get signed up, they can authorize the actions in the application.

4. USER MODULE:

The Entire User Module can be split into 4 distinct parts.

APPOINTMENTS:

The first is the deviant when it comes to the platform of Automobile Servicing, which is Booking an appointment based on your schedule with the whatever workshop the user chooses, just like Zomato or Uber Eats connects customers with a wide range of restaurants.

Our database contains the details of all the workshops who have registered with our application. The appointment will be verified by the workshop and if a slot is free you are notified that you have an appointment fixed.

You also have to choose the servicing mode, with that the employee of that particular workshop will be there on the appointment date to pick your vehicle and drop it back after fixing it up within the appointment time schedule.

If there is no appointment slot free on the precisely entered date and time by user, the user will be notified of that as well. The user can also cancel their appointment.

ASSIST:

The next important function is that, when a user gets into some kind of trouble, say like an engine bust, in the roadside, all he has to do is just load the application and request for backup, and this will choose the most closest workshop and sends the assist request, and based on the location of the workshop, an estimated time is put forth to the user.

E-COMMERCE:

You can also order spare parts that make up any automobile, and the order will again be delivered based on the location of yours and the location of the nearest workshop. If the nearest Workshop does not have the spare parts that you are in requirement of, the application chooses the next nearest workshop.

VIEW HISTORY:

The side segment is the functionality to view your entire history, be it the orders you do, the appointments or the assists that you took up in the entire time you have been a participant of our application.

You can also View your General Profile, edit it, with the inclusion of changing password.

5. ADMIN PROFILE:

Admin will be having their unique ID's and Password taken up from the registration table and connected with the user as specified in the user module. Again, each Admin is actually a workshop.

After logging in,

VIEW/UPDATE PROFILE DETAILS:

Admin will be able to View all his Profile details. He will also be able to Edit/Update his location based on which he will be getting Appointments and Immediate Assist request.

ACKNOWLEDGING THE IMMEDIATE ASSISTANCE FROM THE USER:

Admin will be shown the table in which the details from the user regarding Assistance is stored . He will be able to do the Following:

- Accept the Assistance
- If accepted Estimate time to reach the user's location will be sent to the User.
- Decline the Assist

ACKNOWLEDGING THE BOKING APPOINTMENTS FROM THE USER:

Admin has the rights to view the appointments table,

He will be able to do the Following

Accept the Booking:

- If accepted a notification will be sent to User from Admin

Decline the Booking:

- If declined, the reason for Declining the user appointment will be sent to User.

VIEW THE PARTS ORDERED AND RESPOND ACCORDINGLY:

Table displaying the Details of parts Ordered with quantity will be shown to Admin Admin will be responding (response in the sense placed, in transit or delivered) to the user until item delivery

Incase if user cancels the order, Admin will be updating those details to the Order Table.

GENERATE INVOICE:

- Admin will be able to generate the Invoice of the Vehicle after the service is has a status of completion.
- He can view all the invoices generated and refresh the table to get added invoice.

Maintain All Customer Details:

- View all the customers and their details, also the admin can block certain customers.

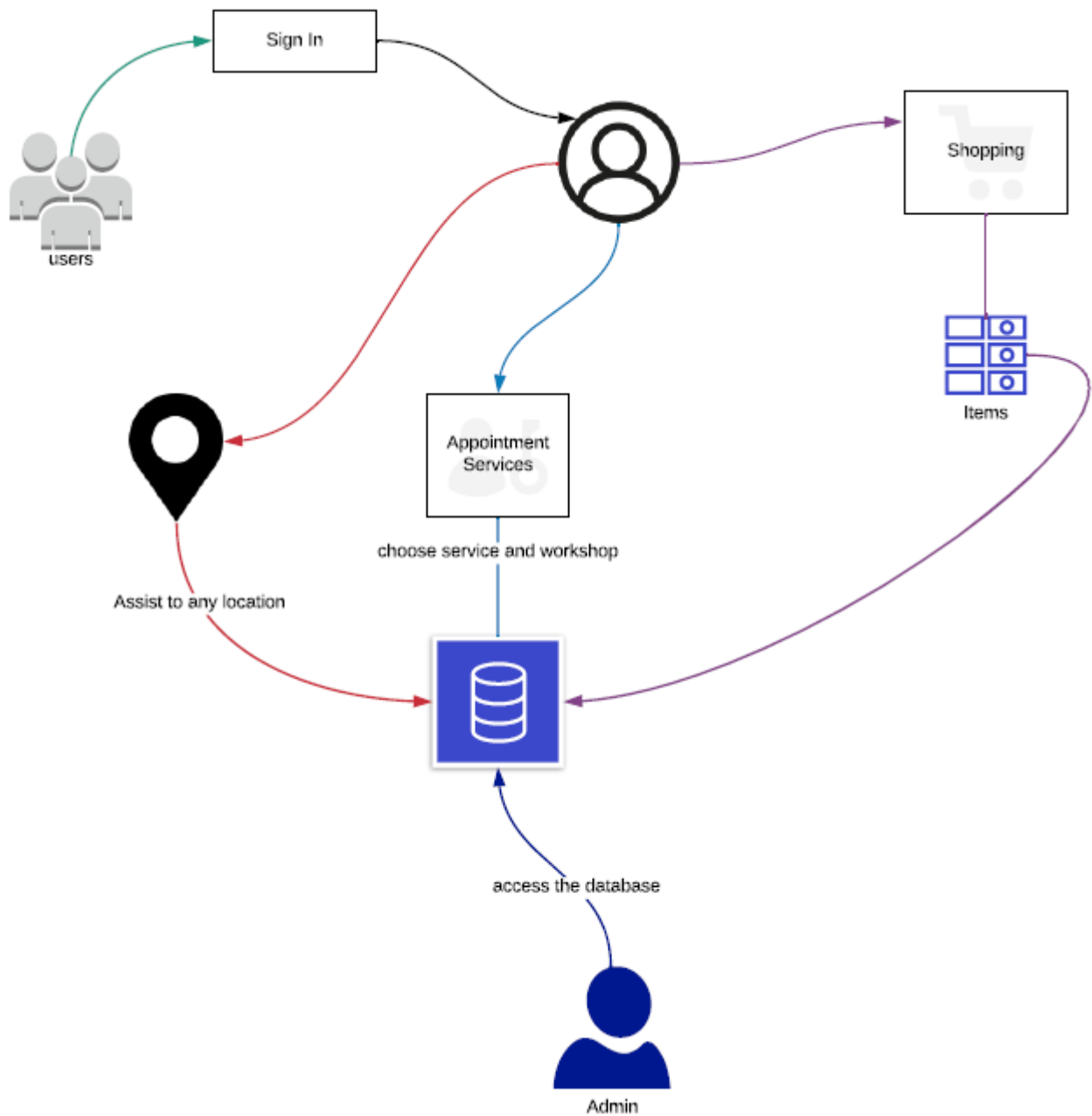
Maintain the Delivery Details:

- A table will be displayed having the details of the vehicle whose work is done and is to be delivered.

When Bill is generated for a vehicle, that particular vehicle's detail will be moved to Delivery Table.

LOW LEVEL DOCUMENTATION

✓ CONTROL FLOW DIAGRAM



✓ ALGORITHM

USER MODULE

Home Screen module:

- i. Start
- ii. Display **user** and **register** button in the **Home Screen module**.
- iii. If the user presses the **register button**:
Then go to the **Register page**.
- iv. If the user presses the **Login button**:
Then go to the **Login page**.
- v. Stop

Register module:

- i. Start
- ii. Connect to the **MySQL Database**
- iii. Use **Users** table in the Database
- iv. Input the User details (name,username,password,mobile no,address line 1,address line 2,district,state,pincode)
- v. Check if the username already exists or not,
If the username already exists,
Display an error message "user already exists".
else
Store the user details in the **Users** table.
Then go to the **Home Screen**.
- vi. Stop

Login module:

- i. Start
- ii. Connect to the **MySQL Database**
- iii. Input the username and password from the user.
- iv. Check whether the username and password exists in the **Users** table
 - If the username and password exists in the **Users** table
Then go to Dashboard.
 - Else
Display the message "Username or Password is incorrect."
- v. Stop

Dashboard Module:

- i. Start
- ii. Connect to the **MySQL Database**
- iii. Display all the user details corresponding to his username from the **Users** table in the database.
- iv. Display all his present assists and appointments corresponding to his username from the **Assists** table and **Appointments** table in the database.
- v. Display all the services provided by the admin
- vi. If the user presses order items button,
 - Display **order items page** in the dashboard.
 - If the user presses schedule button,
Display **schedule items page** in the dashboard.
 - If the user presses cancel button,
Display **cancel Assist and Appointment page** in the dashboard.
 - If the user presses view order button,
Display **view order page** in the dashboard.
 - If the user presses logout button,
Logout from the dashboard and go to the **home screen page**.
- vii. Stop

Order Items Module:

- i. Start
- ii. Display all the items from the **Items table** in the database along with their stock and prize per product value.
- iii. Select the item from the items list.
- iv. Input the quantity required.
- v. If the quantity > stock,
 Display the message "No stock available".
 Else,
 Add the item and quantity to the cart.
- vi. Repeat step 3, step 4 and step 5 to add more items in the cart until the cart is full.
- vii. If the cart is full,
 Display the message "No more items can be added to the cart".
- viii. If the user presses the place order,
 Store all the details in the cart in the **Ecommerce** table.
 Reduce the stock of the items in the **Items** table.
- ix. Stop

Schedule module:

- i. Start
- ii. Display all the functions provided in the schedule module.
- iii. If the user wants- to place an immediate response,
- iv. Input the details (service, vehicle brand, vehicle model, location and pincode).
- v. If the user wants to place an appointment,
 Input the details (vehicle no, vehicle brand , vehicle model , workshop , service , date , time)
- vi. If the user presses the response button,
 Store the details (service, vehicle brand, vehicle model, location and pincode) in the **Assist** table.
 Check the nearby workshop and assign a worker to the user.
- vii. If the user presses the appointment button,

Check whether the time slot is already booked

If the slot is already booked,

Ask the user to choose another time slot

else

Store the details (vehicle no, vehicle brand , vehicle model , workshop , service , date , time) in the **Appointment** table.

viii. Stop

Cancel Assist and Appointment Module:

i. Start

ii. To cancel the assist response,

Select the assist id

Input the reason for the cancellation.

Press cancel button to cancel the assist.

iii. To cancel the appointment,

Select the appointment id

Input the reason for the cancellation.

Press cancel button to cancel the appointment.

iv. Stop

View order Module:

i. Start

ii. Display all the options provided in the View order module

iii. Display all the items placed by the user.

iv. If the user wants to cancel the item placed,

Select the order id

Select the item to be cancelled

Press cancel button to cancel the order.

v. Stop

ADMIN MODULE:

Home Screen module:

- i. Start
- ii. Display **Login** and **register** button in the **Home Screen module**.
- iii. If the user presses the **login button**:
Then go to the **Login page**.
- iv. If the user presses the **Register button**:
Then go to the **Register page**.
- v. Stop

Register module:

- i. Start
- ii. Open JFrame
- iii. Input the User details (workshop name, workshop username, password, mobile number, address) from JTextField.
- iv. Check if the workshop username is already registered or not in the database
If workshop username is already registered,
Then open the login page
Else
Open signup page
- v. Stop

Login module:

- i. Start
- ii. Connect to the **MySQL Database**
- iii. Input the Admin name and password from the admin.
- iv. Check whether the Admin name and password exists in the **Admin** table
If the Admin name and password exists in the **Admin** table
Then go to Admin page.
Else
Display the message "Admin name or Password is incorrect."
- v. Stop

Immediate assistance tab:

- i. Start.
- ii. Assist table is Displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. Admin acknowledges the user.
- v. If "Accept" button is selected,
5.1. Assist is approved.
5.2. Time to reach the location is updated.
- vi. If "Decline" button is selected,
Assist is declined.
- vii. Stop.

View booking tab:

- i. Start.
- ii. Appointments table is displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. Admin acknowledges the user.
- v. If "Accept" button is selected,
5.1. Appointment is confirmed.
5.2. Status of that Appointment is updated.
- vi. If "Decline" button is selected,
Appointment is declined.
- vii. Stop.

Parts required tab:

- i. Start.
- ii. Ecommerce table is displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. Admin updates the status of the order placed.
- v. If admin needs to update status,
Order ID is taken as input and status is updated when "Update Status"
button is selected.
- vi. Stop.

Invoice tab:

- i. Start.
- ii. Details are taken as input.
- iii. If "Generate Invoice" button is selected,
 - 3.1. The status is updated as "Delivered" in Appointments table.
 - 3.2. Invoice is to be generated for the entered Appointment ID.
- iv. If "Cancel" button is selected,
The details entered will be erased.
- v. Stop.

View all invoice tab:

- i. Start.
- ii. All generated invoice by the respective admin will be displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. Stop.

Manage customer details tab:

- i. Start.
- ii. Users table is displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. If "Block User" button is selected,
Status of the particular user is set to "Blocked".
- v. Stop.

View delivery details tab:

- i. Start.
- ii. If status in the appointments table is "Delivered",
That is displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. Stop.

Profile tab:

- i. Start.
- ii. The details of the admin logged in are displayed.
- iii. If "Log out" button is selected,
Admin is logged out.
- iv. If "Refresh All" button is selected,
The entire admin profile is refreshed.
- v. If "Edit Profile" button is selected,
The details are enabled to editable mode.
- vi. If "Submit" button is selected,
The edited changes are saved.
- vii. If "Change Password" button is selected,
The password is changed only after the verification of current password
and when the new password is valid.
- viii. Stop.

Workers tab:

- i. Start.
- ii. Workers table is displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. If the admin needs to change the status,
Worker ID is taken as input and status is updated when "Change Status" is selected.
- v. If "Add Worker" button is selected,
The details of the worker are taken as input and stored.
- vi. Stop.

Stock tab:

- i. Start.
- ii. Items table is displayed.
- iii. If "REFRESH" button is selected,
The Table is refreshed.
- iv. If the admin needs to update the stock,
Product ID is taken as input and stock is updated when "Update Stock" is selected.
- v. If "Add Items" button is selected,
The details of the item are taken as input and stored.
- vi. Stop.

✓ PSEUDO CODE:

USER MODULE

Register module:

Begin

 Connect to the **MySQL Database**
 Use **Users** table in the Database
 Get the User details
 if(username==username in database),
 Print an error message “user already exists”.
 else
 Store the user details in the **Users**

Disconnect the MySQL connection
Redirect to the **Home Screen**.
End

Login module:

Begin

 Connect to the **MySQL Database**
 Get the username and password from the user.
 if(username,password==username,password in the db)
 then go to Dashboard
 else
 Print the error message “Username or Password is incorrect”

Disconnect the MySQL connection.
End

Dashboard Module:

Begin

Display all the user details corresponding to his username

If “dashboard” option is pressed:

Display the user profile, user assist and user appointments details.

If “order items” option is pressed:

Go to order items page.

If “schedule” option is pressed:

Go to schedule page.

If “cancel” option is pressed:

Go to cancel assist and appointments page.

If “view order” option is pressed:

Go to view order.

End

Order Items Module:

Begin

Connect the MySQL database

Display all the items from the **Items table** in the database along with their stock and prize per product value.

total=0;

if(item selected) :

Get quantity required.

if(quantity> stock):

Print “No stock available”

else if(cart==5):

Print “cart is full”

Else:

Insert item,quantity,price in cart

total=total + (quantity * price)

```
If(item cancelled in cart) :  
    total=total-(quantity * price)  
    Remove from cart  
    cart=cart-1  
If “place order” is pressed :  
    Store the order details in the Ecommerce table.  
    stock=stock - quantity for all products placed
```

```
Disconnect the MySql database.  
End.
```

Schedule module :

```
Begin
```

```
    Connect to MySql database
```

```
    If “assist” is needed:
```

```
        Get the details (service,vehicle brand,vehicle model,location and  
        pincode).
```

```
        workshop_min_distance=workshop_distance[0];
```

```
        For i=0 to workshops-1 do:
```

```
            If workshop_min_distance>workshop_distance[i]:
```

```
                Assign the workshop employee to the user.
```

```
        Store all the details in the Assist table.
```

```
    If “appointment” is needed:
```

```
        Get the details (vehicle no, vehicle brand , vehicle model ,  
        workshop , service , date , time)
```

```
        if(time,date==time,date in Appointment table in database):
```

```
            Print(“time slot already booked by someone”)
```

```
        Else:
```

```
            Store the details
```

```
Disconnect the MySQL database
```

```
End
```

Cancel Assist and Appointment Module:

```
Begin
    Connect to the MySQL database;
    If "cancel assist":
        Select assist id
        Get the reason for the cancellation.
        Change the status of assist id as cancelled.

    If "cancel appointment" :
        Select appointment id
        Get the reason for the cancellation.
        Change the status of appointment id as cancelled.
    Disconnect the MySql database.
End
```

View order Module:

```
Begin
    Connect to the MySQL database;
    If "view order" :
        Display all the orders of the users which are not delivered.
    If "cancel order" :
        Select the order id
        Select the product
        Select cancel to cancel the product order by the user.
        Change status of order id as cancelled.
    Disconnect the MySQL database
End
```

ADMIN MODULE

Register module:

Begin

Connect to MySQL database.

Use Admin table in the database

Get details from Admin to register

If (admin name == admin name in database)

Print the error message "Admin already exist"

Else

Save the Admin details in Database

Redirect to Login page

End

Login module:

Begin

Connect to the **MySql Database**

Get the Admin name and password from the user.

If (admin name, password == username, password in the db)

Then go to Admin Page

Else

Print the error message "Username or Password is incorrect"

Disconnect the MySql connection.

End

Immediate Assistance Tab:

```
Begin
    Connect to the MySQL Database
    Assist table is displayed
    If (refresh Button is Selected):
        The entire Table is refreshed, that is database connectivity
        gets refreshed
    If (Accept Assist):
        Assist Id is entered and the related customer's status is
        changed to approved and the worker is sent.
        Set estimated time
    Else"
        The entered assist id is used to change the status to
        declined and the user can see that his request is declined
    Break Connection with Auto-Care Database
End
```

View Booking Tab:

```
Begin
    Connect with MySQL Database
    Appointments Table Gets displayed
    If (refresh):
        The entire table gets refreshed, if database has been
        altered, its show real time

    If (Accept Appointment):
        The particular appointment id is used to pin point the user
        and his requests, the workshop then changes the status to Booked.
        Change status to "OutForPickup" or "OutForDelivery" or
        "VehiclePicked" accordingly
    If (decline Appointment):
        Appointment id is used to change the status to declined and
        his appointment is thereby invalid
    Break Connection with Auto-Care Database
End
```

Parts Required Tab:

Begin

Connectivity to MySql Auto-Care Database is brought up

E Commerce Table is displayed

If (Refresh):

Connectivity to Database Ecommerce table is once again
freshened

If (Update Status):

If (Packed):

Status of that particular order id is changed from
placed to Packed

If (Shipped)

Status of that particular order id is changed to
shipped

If (Delivered):

Status of that particular order id is changed to
delivered

Break Connection with Auto-Care Database

End

Invoice Tab:

Begin

Connectivity with MySql Auto-care Database is established

If Admin enters table

If (Generate Invoice):

Invoice is generated for that appointment id and status of
that appointment is changed to delivery in the appointment tab

Invoice is then added to view all invoices tab

If (cancel):

Entered details are all erased

Break Connection with Auto-Care Database

End

View All Invoices Tab:

Begin

Connectivity with MySql Auto-care Database is established

All the invoices are shown that were generated by that particular Admin

End

Manage Customer Details Tab:

Begin

Establish connection with MySql Auto-Care Database

User Details are shown, all the users registered with Auto-Care.

If (Block User)

Entered user id is used to change the status of that user as "Blocked" and is not let to login from next time.

Break Connection with Auto-Care Database

End

View Delivery Details Tab:

Begin

MySql Auto-Care Database is connected

Delivered vehicle details are shown, that is appointments that were completed

If (refresh):

Auto-Care database connectivity is refreshed to show most recent data

Break Connection with Auto-Care Database

End

Profile Tab:

Begin

Connection with MySql Auto-Care Database is established

Admin details of the particular Admin who has logged in is shown

If (logout):

Redirects out of admin page to main page.

If (refresh all):

The entire program is refreshed

If (Edit Profile):

Edit admin profile as preferred

If (Submit)

Edited profile is updated into the Admin table In the

database

If (change password)

The password is changed only after the verification of current password and only when the new password is valid.

Break Connection with Auto-Care Database

End

Workers Tab:

Begin

Connection with MySql Auto-Care Database is tied

Workers table is displayed

if(Refresh)

Current table attributes are shown

if(change status):

You can change status of a worker from free to not free and back again

if(Add Worker):

worker name and worker number are used to update the workers table with their details

Break Connection with Auto-Care Database

End

Stock Tab:

Connection with MySql Auto-Care Database is brought up

Items table is displayed

If (Refresh)

Current table, that is once when the admin has logged in and wants to see changes made after his login, attributes are shown

If (Update Stock):

You can change stock values of an item using its id

If (Add Items):

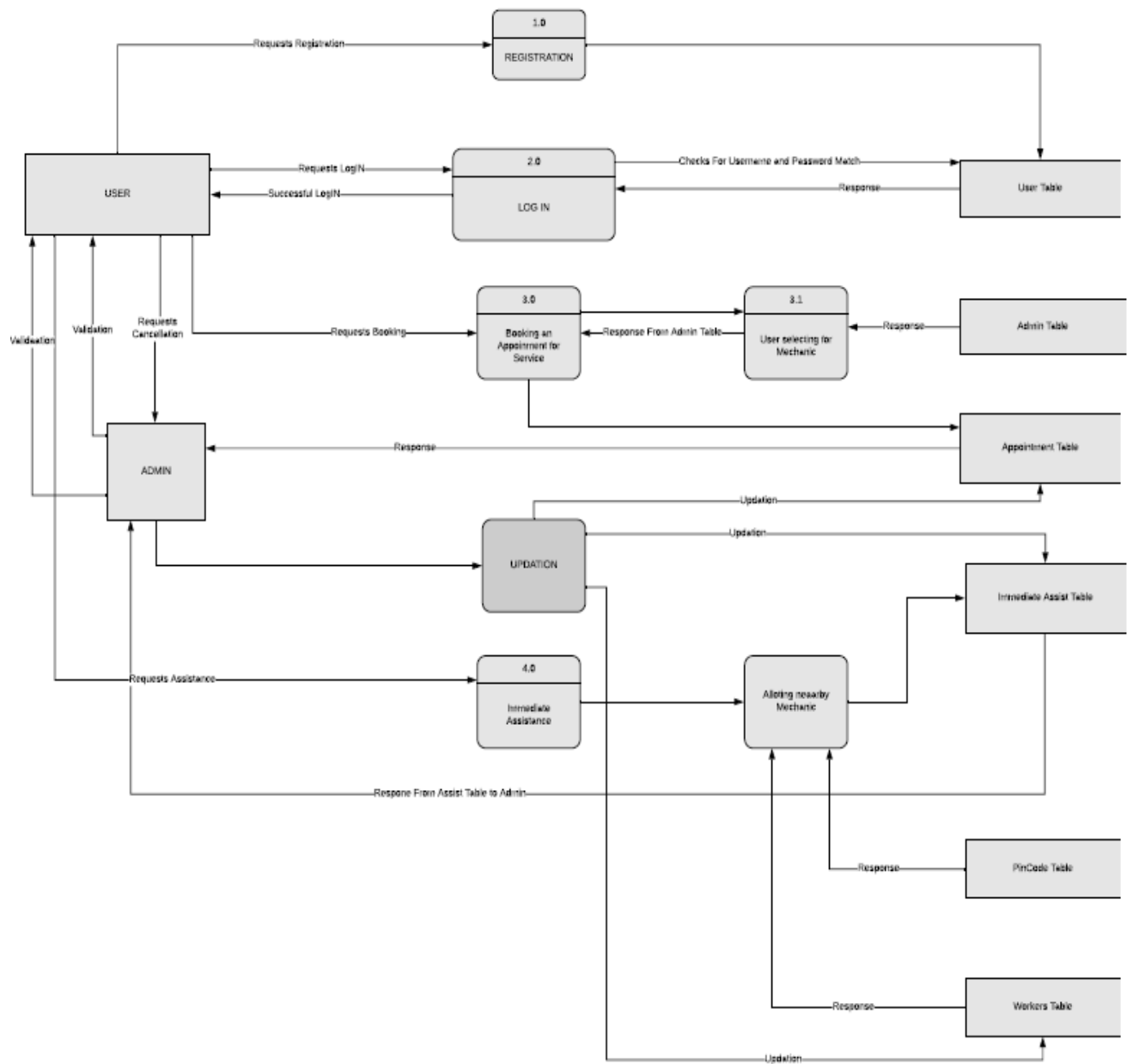
Item name and item price are used to update the items table with their details

Break Connection with Auto-Care Database

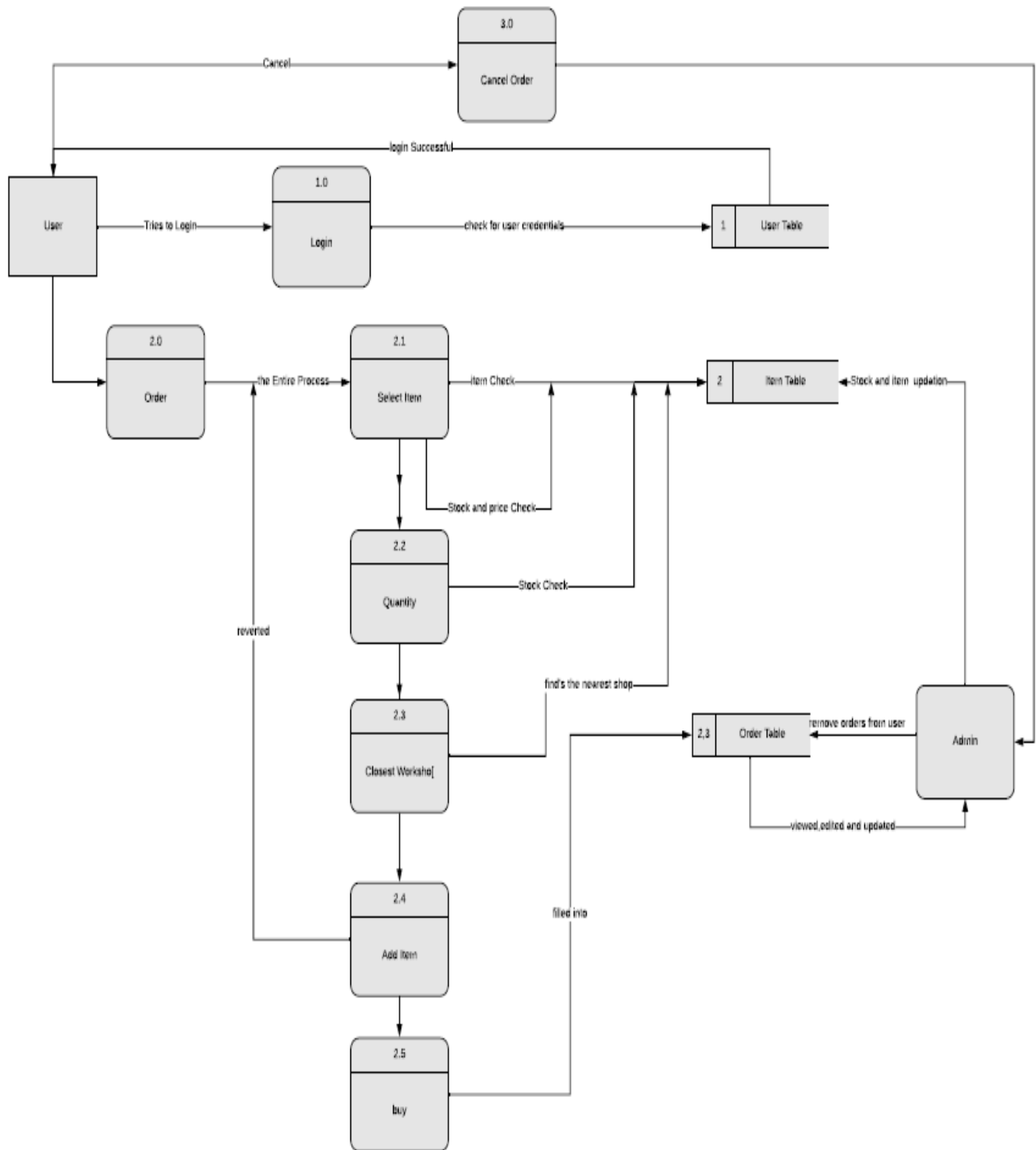
End

✓ DATA FLOW DIAGRAM

FOR APPOINTMENTS AND ASSISTS:



E-COMMERCE



SOURCE CODE

USER MODULE

ProjectMain.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;

/**
 *
 * @author PROJECTS
 */
public class ProjectMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Screen s=new Screen();
        s.scr();
    }

}
```

Screen.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.RoundRectangle2D;
```



```

import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
/**
 *
 * @author PROJECTS
 */
public class Screen {
    public void scr(){
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setUndecorated(true);

        double x = 0;
        double y = 0;
        frame.setShape(new RoundRectangle2D.Double(x, y, 1200, 600, 50, 50));
        frame.setSize(1200, 600);
        frame.setBounds(30, 30, 1200, 600);
        frame.getContentPane().setBackground(Color.white);

        Icon icon = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img3.png");
        JButton log = new JButton(icon);
        log.setBorderPainted(false);
        log.setFocusPainted(false);
        log.setContentAreaFilled(false);
        log.setBounds(790, 215, 250, 100);
        frame.add(log);

        Icon icon1 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img4.png");
        JButton reg = new JButton(icon1);
        reg.setBorderPainted(false);
        reg.setFocusPainted(false);
        reg.setContentAreaFilled(false);
        reg.setBounds(790, 380, 250, 100);
        frame.add(reg);
    }
}

```

```

log.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        login l=new login();
        l.log();
        frame.setVisible(false);
    }
});

reg.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        Register r=new Register();
        r.reg();
        frame.setVisible(false);
    }
});

ImageIcon iconss = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img1.png");
JLabel labels = new JLabel(iconss);
labels.setBounds(10,70,500,500);
frame.add(labels);

ImageIcon icons = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img2.png");
JLabel label = new JLabel(icons);
label.setBounds(550,-15,700,600);
frame.add(label);

frame.setLayout(null);
frame.setVisible(true);

}
}

```

Register.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;

import java.awt.Color;
import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.geom.RoundRectangle2D;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;

/**
 *
 * @author PROJECTS
 */
public class Register {
    JFrame frame;
    String opt="coimbatore";
    public void reg(){
```

```
frame = new JFrame();
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setUndecorated(true);

double x = 0;
double y = 0;
frame.setShape(new RoundRectangle2D.Double(x, y, 1200, 600, 50, 50));
frame.setSize(1200, 600);
frame.setBounds(30, 30, 1200, 600);
frame.getContentPane().setBackground(Color.white);

ImageIcon carimg = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img1.png");
JLabel labelcar = new JLabel(carimg);
labelcar.setBounds(10, 70, 500, 500);

ImageIcon regimg = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img5.png");
JLabel labelreg = new JLabel(regimg);
labelreg.setBounds(500, -20, 700, 600);

Icon registerbtnimg = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img6.png");
JButton registerbtn = new JButton(registerbtnimg);
registerbtn.setBorderPainted(false);
registerbtn.setFocusPainted(false);
registerbtn.setContentAreaFilled(false);
registerbtn.setBounds(920, 520, 200, 100);

JPanel panelback = new JPanel();
panelback.setLayout(null);
panelback.setBackground(Color.white);
panelback.setBounds(0, 0, 1200, 600);

ArrayList<String> statelist = new ArrayList<String>();
try{
```

```

        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
        Statement st=conn.createStatement();
        String sql = "SELECT state from state;";
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            statelist.add(rs.getString("state"));

        }
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
    String[] statearr=new String[statelist.size()];
    statelist.toArray(statearr);
    JComboBox state=new JComboBox(statearr);
    state.setBounds(904,306,240,30);
    state.setBackground(Color.white);
    JPanel panel=new JPanel();
    panel.setBounds(904,390,240,40);
    panel.setLayout(null);
    panel.setBackground(Color.white);

state.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools | Templates.
        Component[] components=panel.getComponents();
        for(Component component :components){

            panel.remove(component);
        }
        panel.revalidate();
        panel.repaint();
        String inputdistrict=state.getSelectedItem().toString();
        panel.setVisible(false);
        ArrayList<String> districtlist=new ArrayList<String>();

```

```

try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

    Statement st=conn.createStatement();
    String sql = "SELECT district from district where state='"+inputdistrict+"";
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()){
        districtlist.add(rs.getString("district"));
    }
    System.out.println("sucess district");
    System.out.println(districtlist);
    conn.close();
}
catch(Exception e){
    System.out.println(" "+e);
}
String[] districtarr=new String[districtlist.size()];
districtlist.toArray(districtarr);
JComboBox district=new JComboBox(districtarr);
district.setBounds(0,0,210,30);
district.setBackground(Color.white);

district.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools |
Templates.
        opt=district.getSelectedItem().toString();
        System.out.println(opt);
    }
});

panel.setBackground(Color.white);
panel.add(district);
panelback.updateUI();
panel.setVisible(true);
panel.updateUI();

```

```
    }  
});  
panelback.add(state);  
  
JTextField tf1=new JTextField();  
JTextField tf2=new JTextField();  
JPasswordField tf3=new JPasswordField();  
JPasswordField tf4=new JPasswordField();  
JTextField tf5=new JTextField();  
JTextField tf6=new JTextField();  
JTextField tf7=new JTextField();  
JTextField tf8=new JTextField();  
  
tf1.setBounds(600,227,210,30);  
tf2.setBounds(600,303,210,30);  
tf3.setBounds(540,385,100,30);  
tf4.setBounds(700,385,100,30);  
tf5.setBounds(560,471,240,30);  
tf6.setBounds(904,142,240,30);  
tf7.setBounds(904,227,240,30);  
tf8.setBounds(904,471,240,30);  
  
tf1.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf2.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf3.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf4.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf5.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf6.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf7.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
tf8.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
  
tf1.setBackground(new Color(240,240,240));  
tf2.setBackground(new Color(240,240,240));  
tf3.setBackground(new Color(240,240,240));  
tf4.setBackground(new Color(240,240,240));  
tf5.setBackground(new Color(240,240,240));  
tf6.setBackground(new Color(240,240,240));  
tf7.setBackground(new Color(240,240,240));  
tf8.setBackground(new Color(240,240,240));
```

```
panelback.add(tf1);
panelback.add(tf2);
panelback.add(tf3);
panelback.add(tf4);
panelback.add(tf5);
panelback.add(tf6);
panelback.add(tf7);
panelback.add(tf8);
```

```
panelback.add(labelcar);
panelback.add(labelreg);
panelback.add(registerbtn);
```

```
registerbtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        Pattern p1=Pattern.compile("[^A-Za-z]");
        Matcher m1=p1.matcher(tf1.getText().toString());
        boolean tf1b=m1.find();

        Pattern p2=Pattern.compile("[^0-9]");
        Matcher m2=p2.matcher(tf5.getText().toString());
        boolean tf5b=m2.find();

        Pattern p5=Pattern.compile("[^0-9]");
        Matcher m5=p5.matcher(tf5.getText().toString());
        boolean tf8b=m2.find();

        boolean error=false;
        if(tf1b){
            Error e=new Error();
            e.errordis("Name cannot contain special character","Pls change it");
            error=true;
        }
    }
});
```



```

else if(tf5b){
    Error e=new Error();
    e.errordis("Mobilenos can contain only digits","Pls change it");
    error=true;
}
else if(tf5.getText().toString().length()!=10){
    Error e=new Error();
    e.errordis("Mobilenos should be 10 digits","Pls change it");
    error=true;
}

else if(tf8b){
    Error e=new Error();
    e.errordis("Pincode can contain only numbers","Pls change it");
}
else if(tf8.getText().toString().length()!=6){
    Error e=new Error();
    e.errordis("Pincode should be 6 digits","Pls change it");
    error=true;
}
else if(!(new String(tf3.getPassword()).equals(new
String(tf4.getPassword())))){
    Error e=new Error();
    e.errordis("Passwords are not matched","Check it");
    error=true;
}
else
if(tf1.getText().isEmpty() || tf2.getText().isEmpty() || tf3.getPassword().toString().isEmpty() || tf4.getPassword().toString().isEmpty() || tf5.getText().isEmpty() || tf6.getText().isEmpty() || tf7.getText().isEmpty() || tf8.getText().isEmpty()){
    Error e=new Error();
    e.errordis("Enter all fields","Check it");
    error=true;
}
//else if()
// {

//}
else{
    boolean empty=true;
    try{

```

```

        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String sql = "select username from users where
username='"+tf2.getText().toString()+"'";
        ResultSet rs  = st.executeQuery(sql);
        while(rs.next()){
            empty=false;
            System.out.println("alreadyexist");
        }
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
    if(!empty){
        Error e=new Error();
        e.errorid("Username already exists","Pls change it");
        error=true;
    }
}
if(error==false){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        String pass=new String(tf3.getPassword());
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String s="insert into
users(name,username,password,mobileno,address1,address2,district,state,pincode,s
tatus)
values('"+tf1.getText().toString()+"','"+tf2.getText().toString()+"','"+pass+"','"+tf5.get
Text().toString()+"','"+tf6.getText().toString()+"','"+tf7.getText().toString()+"','"+opt+"
','"+state.getSelectedItem().toString()+"','"+tf8.getText().toString()+"','active')";
        st.executeUpdate(s);
        System.out.println("Success!");
        messagebox m=new messagebox();
        m.message();
    }
}

```

```

        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
}

}

});

```

```

frame.add(panel);
frame.add(panelback);

```

```

frame.setLayout(null);
frame.setVisible(true);

```

```

}
public static void main(String[] args){
    Register r=new Register();
    r.reg();
}

```

```

public class messagebox {

```

```

    public void message(){

```

```

        JFrame frame1 = new JFrame();
        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame1.setUndecorated(true);

```

```

        double x = 0;
        double y = 0;
        frame1.setShape(new RoundRectangle2D.Double(x, y,400,200, 50, 50));
        frame1.setSize(400, 200);
        frame1.setBounds(600,250,400,200);
        frame1.getContentPane().setBackground(new Color(160,237,240));

```

```
Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img8.png");  
JButton b = new JButton/icons);  
b.setBorderPainted(false);  
b.setFocusPainted(false);  
b.setContentAreaFilled(false);  
b.setBounds(190,70,200,100);  
frame1.add(b);
```

```
ImageIcon icon = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img7.png");  
JLabel labels = new JLabel(icon);  
labels.setBounds(3,15,350,150);  
frame1.add(labels);
```

```
b.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        Screen s=new Screen();  
        s.scr();  
        frame1.setVisible(false);  
        frame.setVisible(false);  
    }  
});
```

```
frame1.setLayout(null);  
frame1.setVisible(true);  
}  
  
}  
  
}
```

login.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;

/**
 *
 * @author PROJECTS
 */
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.RoundRectangle2D;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
import javax.swing.JPasswordField;
/**
 *
 * @author PROJECTS
 */
public class login {
    public void log(){
        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setUndecorated(true);
```

```
double x = 0;
double y = 0;
frame.setShape(new RoundRectangle2D.Double(x, y, 1200, 600, 50, 50));
frame.setSize(1200, 600);
frame.setBounds(30, 30, 1200, 600);
frame.getContentPane().setBackground(Color.white);

Icon icon = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img10.png");
JButton b = new JButton(icon);
b.setBorderPainted(false);
b.setFocusPainted(false);
b.setContentAreaFilled(false);
b.setBounds(750, 400, 250, 100);
frame.add(b);

JTextField tf1 = new JTextField();
JPasswordField tf2 = new JPasswordField();

tf1.setBounds(820, 220, 210, 50);
tf2.setBounds(730, 323, 300, 50);

tf1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
tf2.setBorder(javax.swing.BorderFactory.createEmptyBorder());

tf1.setBackground(new Color(217, 217, 217));
tf2.setBackground(new Color(217, 217, 217));

frame.add(tf1);
frame.add(tf2);

ImageIcon iconss = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img1.png");
JLabel labels = new JLabel(iconss);
```

```

labels.setBounds(10,70,500,500);
frame.add(labels);

ImageIcon icons = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img9.png");
JLabel label = new JLabel(icons);
label.setBounds(650,-25,500,500);
frame.add(label);

b.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        try
        {
            Class.forName("com.mysql.jdbc.Driver");

            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st=conn.createStatement();
            String pass=new String(tf2.getPassword());
            String sql = "select userid,username,password,status from users
where username='"+tf1.getText().toString()+"' and password='"+pass+"'";
            ResultSet rs  = st.executeQuery(sql);
            // loop through the result set

            boolean success=false;
            int userid=0;int flag=0;
            while(rs.next())
            {
                userid=rs.getInt("userid");
                if(rs.getString("status").equals("Blocked"))
                {
                    Error w=new Error();
                    w.errordis("You are Blocked from entry","");
                    flag=1;
                }
                else
                {

```

```

        success=true;
    }
}

if(success==true){

    dashboard d=new dashboard();
    d.dash(userid);
    frame.setVisible(false);
    System.out.println("loginsuccessful");
}
else if(success==false && flag==0)
{
    Error e=new Error();
    e.erroris("Login Unsuccessful","Check entries !!! retry
again!!!");

    System.out.println("not successful");
}
else
{

}

conn.close();

}
catch(Exception e)
{
    System.out.println("    "+e);
}

}
});

frame.setLayout(null);
frame.setVisible(true);

}
}

```


Dashboard.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;

/**
 *
 * @author PROJECTS
 */
import java.sql.Timestamp;
import java.awt.Color;
import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.geom.RoundRectangle2D;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.time.Instant;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.swing.JCheckBox;
import javax.swing.JTextArea;

/**
```

```

*
* @author PROJECTS
*/
public class dashboard {

    String serviceinput="";
    String dateinput="";
    String timeinput="";
    String vehmodelinput="";

    String vehmodelinputass="";
    int totalitemscart=0;
    String productnameinputcan="";
    public void dash(int userid){

        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setUndecorated(true);

        double x = 0;
        double y = 0;
        frame.setShape(new RoundRectangle2D.Double(x, y,1200,600, 50, 50));
        frame.setSize(1200, 600);
        frame.setBounds(30,30,1200,600);
        frame.getContentPane().setBackground(Color.white);

        //Design buttons

        Icon icon = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img12.png");
        JButton b = new JButton(icon);
        b.setBorderPainted(false);
        b.setFocusPainted(false);
        b.setContentAreaFilled(false);
        b.setBounds(-50,30,250,100);
        frame.add(b);
    }
}

```

```
Icon icon1 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img13.png");  
JButton b1 = new JButton(icon1);  
b1.setBorderPainted(false);  
b1.setFocusPainted(false);  
b1.setContentAreaFilled(false);  
b1.setBounds(-50,100,250,100);  
frame.add(b1);
```

```
Icon icon2 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img15.png");  
JButton b2= new JButton(icon2);  
b2.setBorderPainted(false);  
b2.setFocusPainted(false);  
b2.setContentAreaFilled(false);  
b2.setBounds(-50,170,250,100);  
frame.add(b2);
```

```
Icon icon3 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img17.png");  
JButton b3 = new JButton(icon3);  
b3.setBorderPainted(false);  
b3.setFocusPainted(false);  
b3.setContentAreaFilled(false);  
b3.setBounds(-50,240,250,100);  
frame.add(b3);
```

```
Icon icon8 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img44.png");  
JButton b4 = new JButton(icon8);  
b4.setBorderPainted(false);  
b4.setFocusPainted(false);  
b4.setContentAreaFilled(false);  
b4.setBounds(-50,310,250,100);  
frame.add(b4);
```

```
Icon icon10 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img49.png");  
JButton b5 = new JButton(icon10);  
b5.setBorderPainted(false);
```

```
b5.setFocusPainted(false);
b5.setContentAreaFilled(false);
b5.setBounds(-50,500,250,100);
frame.add(b5);
```

```
b5.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        frame.setVisible(false);
        Screen s=new Screen();
        s.scr();

    }
});
```

```
ImageIcon icons = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img19.png");
JLabel label = new JLabel(icons);
label.setBounds(0,0,150,800);
frame.add(label);
```

```
Icon icon4 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img11.png");
Icon icon5 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img14.png");
Icon icon6 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img16.png");
Icon icon7 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img18.png");
Icon icon9 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img45.png");
```

```
//Dashboard start
JPanel dashboard=new JPanel();
```

```

dashboard.setBackground(Color.white);

dashboard.setLayout(null);
dashboard.setBounds(150,0,1150,600);

Imagelcon iconprofile = new
Imagelcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img29.png");
JLabel labelprofile = new JLabel(iconprofile);
labelprofile.setBounds(-50,10,700,500);

Imagelcon icondashapp = new
Imagelcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img31.png");
JLabel labeldashapp = new JLabel(icondashapp);
labeldashapp.setBounds(450,-80,700,500);

Imagelcon icondaass = new
Imagelcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img30.png");
JLabel labeldashass = new JLabel(icondaass);
labeldashass.setBounds(450,150,700,500);

try
{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
Statement st=conn.createStatement();
String sql = "SELECT
username,mobileno,name,address1,address2,district,state,pincode FROM users
where userid='"+userid+"'";
ResultSet rs = st.executeQuery(sql);
JLabel username;
JLabel mobileno;
JLabel name;
JLabel address1;
JLabel address2;
JLabel district;
JLabel state;
JLabel pincode;

```

```
while(rs.next()){

    username=new JLabel(rs.getString("username"));
    username.setBounds(320,8,200,70);
    username.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    username.setBackground(new Color(217,217,217));
    dashboard.add(username);

    mobileno=new JLabel(rs.getString("mobilenno"));
    mobileno.setBounds(320,60,200,70);
    mobileno.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    mobileno.setBackground(new Color(217,217,217));
    dashboard.add(mobilenno);

    name=new JLabel(rs.getString("name"));
    name.setBounds(210,140,200,70);
    name.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    name.setBackground(new Color(217,217,217));
    dashboard.add(name);

    address1=new JLabel(rs.getString("a d d r e s s 1"));
    address1.setBounds(210,200,200,70);
    address1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    address1.setBackground(new Color(217,217,217));
    dashboard.add(address1);

    address2=new JLabel(rs.getString("a d d r e s s 2"));
    address2.setBounds(210,240,200,70);
    address2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    address2.setBackground(new Color(217,217,217));
    dashboard.add(address2);

    district=new JLabel(rs.getString("district"));
    district.setBounds(110,330,200,70);
    district.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    district.setBackground(new Color(217,217,217));
    dashboard.add(district);

    state=new JLabel(rs.getString("state"));
    state.setBounds(350,330,200,70);
```

```
state.setBorder(javax.swing.BorderFactory.createEmptyBorder());
state.setBackground(new Color(217,217,217));
dashboard.add(state);

pincode=new JLabel(rs.getString("pincode"));
pincode.setBounds(110,420,200,70);
pincode.setBorder(javax.swing.BorderFactory.createEmptyBorder());
pincode.setBackground(new Color(217,217,217));
dashboard.add(pincode);
```

```
}
JTextArea apptablet1;
JTextArea apptablet2;
JTextArea apptablet3;
JTextArea apptablet4;
```

```
String s = "SELECT appointmentid,name,date,id from appointments a JOIN admin
ad ON a.adminid=ad.adminid where userid='"+userid+"' and status!='Declined' and
status!='cancelled' and status!='OutForDelivery'";
```

```
ResultSet r = st.executeQuery(s);
String t1="";
String t2="";
String t3="";
String t4="";
```

```
apptablet1=new JTextArea("");
apptablet1.setBounds(600,120,20,100);
apptablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet1.setBackground(new Color(255,230,133));
apptablet1.setEditable(false);
```

```
apptablet2=new JTextArea("");
apptablet2.setBounds(655,120,150,100);
apptablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet2.setBackground(new Color(255,230,133));
apptablet2.setEditable(false);
```

```
apptablet3=new JTextArea("");
apptablet3.setBounds(850,120,70,100);
apptablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet3.setBackground(new Color(255,230,133));
apptablet3.setEditable(false);
```

```
apptablet4=new JTextArea("");
apptablet4.setBounds(950,120,70,100);
apptablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet4.setBackground(new Color(255,230,133));
apptablet4.setEditable(false);
```

```
while(r.next()){
    t1=t1+"\n"+r.getString("appointmentid");
    t2=t2+"\n"+r.getString("name");
    t3=t3+"\n"+r.getString("date");
    t4=t4+"\n"+r.getString("id").substring(0,8);
}
apptablet1.setText(t1);
apptablet2.setText(t2);
apptablet3.setText(t3);
apptablet4.setText(t4);
```

```
dashboard.add(apptablet1);
dashboard.add(apptablet2);
dashboard.add(apptablet3);
dashboard.add(apptablet4);
```

```
JTextArea asstablet1;
JTextArea asstablet2;
JTextArea asstablet3;
JTextArea asstablet4;
```



```
String s1 = "select assistid,name,workername,workermobileneno from assist a join  
workers w ON a.workerid=w.workerid join admin ad ON ad.adminid=w.adminid  
where userid='"+userid+"' and status='Approved'";
```

```
ResultSet r1 = st.executeQuery(s1);
```

```
String ta1="";
```

```
String ta2="";
```

```
String ta3="";
```

```
String ta4="";
```

```
asstablet1=new JTextArea("");
```

```
asstablet1.setBounds(600,380,20,50);
```

```
asstablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
```

```
asstablet1.setBackground(new Color(255, 197, 232));
```

```
asstablet1.setEditable(false);
```

```
asstablet2=new JTextArea("");
```

```
asstablet2.setBounds(655,380,150,50);
```

```
asstablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
```

```
asstablet2.setBackground(new Color(255,197,232));
```

```
asstablet2.setEditable(false);
```

```
asstablet3=new JTextArea("");
```

```
asstablet3.setBounds(850,380,70,50);
```

```
asstablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());
```

```
asstablet3.setBackground(new Color(255,197,232));
```

```
asstablet3.setEditable(false);
```

```
asstablet4=new JTextArea("");
```

```
asstablet4.setBounds(950,380,70,50);
```

```
asstablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
```

```
asstablet4.setBackground(new Color(255,197,232));
```

```
asstablet4.setEditable(false);
```

```
while(r1.next()){
```

```
    ta1=ta1+"\n"+r1.getString("assistid");
```

```
    ta2=ta2+"\n"+r1.getString("name");
```

```
    ta3=ta3+"\n"+r1.getString("workername");
```

```

        ta4=ta4+"\n"+r1.getString("workermobileno");
    }
    asstablet1.setText(ta1);
    asstablet2.setText(ta2);
    asstablet3.setText(ta3);
    asstablet4.setText(ta4);

    dashboard.add(asstablet1);
    dashboard.add(asstablet2);
    dashboard.add(asstablet3);
    dashboard.add(asstablet4);

    conn.close();
}
catch(Exception e)
{
    System.out.println(" "+e);
}
Icon iconclose = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img57.png");
JButton close= new JButton(iconclose);
close.setBorderPainted(false);
close.setFocusPainted(false);
close.setContentAreaFilled(false);
close.setBounds(670,500,250,100);

close.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        frame.setVisible(false);
        table d=new table();
        d.tables(userid);

    }
});

dashboard.add(close);

```

```
dashboard.add(label dashass);  
dashboard.add(label dashapp);  
dashboard.add(label profile);
```

```
dashboard.setVisible(true);  
frame.add(dashboard);  
//dashboard end
```

```
//panel order
```

```
JPanel order = new JPanel();  
order.setBackground(Color.white);
```

```
order.setLayout(null);  
order.setBounds(150,0,1150,600);
```

```
order.setVisible(false);  
frame.add(order);
```

```
//panel order end
```

```
//panel schedule
```

```
JPanel schedule = new JPanel();  
schedule.setBackground(Color.white);
```

```
schedule.setLayout(null);  
schedule.setBounds(150,0,1150,600);
```

```
schedule.setVisible(false);  
frame.add(schedule);  
//panel schedule end
```

```
//panel cancel start
```

```
JPanel cancel = new JPanel();  
cancel.setBackground(Color.white);  
cancel.setLayout(null);  
cancel.setBounds(150,0,1150,600);
```

```
cancel.setVisible(false);  
frame.add(cancel);  
//panel cancel end
```

```
//panel vieworder  
JPanel vieworder = new JPanel();  
vieworder.setBackground(Color.white);  
vieworder.setLayout(null);  
vieworder.setBounds(150,0,1150,600);
```

```
vieworder.setVisible(false);  
frame.add(vieworder);
```

```
//panel vieworder end
```

```
b.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        dashboard.removeAll();  
        dashboard.revalidate();  
        dashboard.repaint();  
        dashboard.setVisible(true);  
  
        b.setIcon(icon);  
        b1.setIcon(icon1);
```

```
b2.setIcon(icon2);
b3.setIcon(icon3);
b4.setIcon(icon8);
```

```
ImageIcon iconprofile = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img29.png");
JLabel labelprofile = new JLabel(iconprofile);
labelprofile.setBounds(-50,10,700,500);
```

```
ImageIcon icondashapp = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img31.png");
JLabel labeldashapp = new JLabel(icondashapp);
labeldashapp.setBounds(450,-80,700,500);
```

```
ImageIcon icondaass = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img30.png");
JLabel labeldashass = new JLabel(icondaass);
labeldashass.setBounds(450,150,700,500);
```

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

    Statement st=conn.createStatement();
    String sql = "SELECT
username,mobileno,name,address1,address2,district,state,pincode FROM users
where userid='"+userid+"'";
    ResultSet rs = st.executeQuery(sql);
    JLabel username;
    JLabel mobileno;
    JLabel name;
    JLabel address1;
    JLabel address2;
    JLabel district;
    JLabel state;
    JLabel pincode;

    while(rs.next()){
        username=new JLabel(rs.getString("username"));
```

```
username.setBounds(320,8,200,70);
username.setBorder(javax.swing.BorderFactory.createEmptyBorder());
username.setBackground(new Color(217,217,217));
dashboard.add(username);
```

```
mobilenos=new JLabel(rs.getString("mobilenos"));
mobilenos.setBounds(320,60,200,70);
mobilenos.setBorder(javax.swing.BorderFactory.createEmptyBorder());
mobilenos.setBackground(new Color(217,217,217));
dashboard.add(mobilenos);
```

```
name=new JLabel(rs.getString("name"));
name.setBounds(210,140,200,70);
name.setBorder(javax.swing.BorderFactory.createEmptyBorder());
name.setBackground(new Color(217,217,217));
dashboard.add(name);
```

```
address1=new JLabel(rs.getString("address1"));
address1.setBounds(210,200,200,70);
address1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
address1.setBackground(new Color(217,217,217));
dashboard.add(address1);
```

```
address2=new JLabel(rs.getString("address2"));
address2.setBounds(210,240,200,70);
address2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
address2.setBackground(new Color(217,217,217));
dashboard.add(address2);
```

```
district=new JLabel(rs.getString("district"));
district.setBounds(110,330,200,70);
district.setBorder(javax.swing.BorderFactory.createEmptyBorder());
district.setBackground(new Color(217,217,217));
dashboard.add(district);
```

```
state=new JLabel(rs.getString("state"));
state.setBounds(350,330,200,70);
state.setBorder(javax.swing.BorderFactory.createEmptyBorder());
state.setBackground(new Color(217,217,217));
dashboard.add(state);
```

```

        pincode=new JLabel(rs.getString("pincode"));
        pincode.setBounds(110,420,200,70);
        pincode.setBorder(javax.swing.BorderFactory.createEmptyBorder());
        pincode.setBackground(new Color(217,217,217));
        dashboard.add(pincode);
    }
    JTextArea apptablet1;
    JTextArea apptablet2;
    JTextArea apptablet3;
    JTextArea apptablet4;
    String s = "SELECT appointmentid,name,date,id from appointments a JOIN
admin ad ON a.adminid=ad.adminid where userid='"+userid+"' and
status!='Declined' and status!='cancelled' and status!='OutForDelivery'";
    ResultSet r  = st.executeQuery(s);
    String t1="";
    String t2="";
    String t3="";
    String t4="";

    apptablet1=new JTextArea("");
    apptablet1.setBounds(600,120,20,100);
    apptablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    apptablet1.setBackground(new Color(255,230,133));
    apptablet1.setEditable(false);

    apptablet2=new JTextArea("");
    apptablet2.setBounds(655,120,150,100);
    apptablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    apptablet2.setBackground(new Color(255,230,133));
    apptablet2.setEditable(false);

    apptablet3=new JTextArea("");
    apptablet3.setBounds(850,120,70,100);
    apptablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    apptablet3.setBackground(new Color(255,230,133));
    apptablet3.setEditable(false);

    apptablet4=new JTextArea("");

```

```
apptablet4.setBounds(950,120,70,100);
apptablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet4.setBackground(new Color(255,230,133));
apptablet4.setEditable(false);
```

```
while(r.next()){
    t1=t1+"\n"+r.getString("appointmentid");
    t2=t2+"\n"+r.getString("name");
    t3=t3+"\n"+r.getString("date");
    t4=t4+"\n"+r.getString("id").substring(0,8);
}
apptablet1.setText(t1);
apptablet2.setText(t2);
apptablet3.setText(t3);
apptablet4.setText(t4);
```

```
dashboard.add(apptablet1);
dashboard.add(apptablet2);
dashboard.add(apptablet3);
dashboard.add(apptablet4);
```

```
JTextArea asstablet1;
JTextArea asstablet2;
JTextArea asstablet3;
JTextArea asstablet4;
```

```
String s1 = "select assistid,name,workername,workermobileno from assist
a join workers w ON a.workerid=w.workerid join admin ad ON
ad.adminid=w.adminid where userid='"+userid+"' and status='Approved'";
```

```
ResultSet r1 = st.executeQuery(s1);
String ta1="";
String ta2="";
String ta3="";
String ta4="";
```

```
asstablet1=new JTextArea("");
asstablet1.setBounds(600,380,20,50);
asstablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet1.setBackground(new Color(255, 197, 232));
asstablet1.setEditable(false);
```



```
asstablet2=new JTextArea("");
asstablet2.setBounds(655,380,150,50);
asstablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet2.setBackground(new Color(255,197,232));
asstablet2.setEditable(false);
```

```
asstablet3=new JTextArea("");
asstablet3.setBounds(850,380,70,50);
asstablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet3.setBackground(new Color(255,197,232));
asstablet3.setEditable(false);
```

```
asstablet4=new JTextArea("");
asstablet4.setBounds(950,380,70,50);
asstablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet4.setBackground(new Color(255,197,232));
asstablet4.setEditable(false);
```

```
while(r1.next()){
    ta1=ta1+"\n"+r1.getString("assistid");
    ta2=ta2+"\n"+r1.getString("name");
    ta3=ta3+"\n"+r1.getString("workername");
    ta4=ta4+"\n"+r1.getString("workermobilenos");
}
asstablet1.setText(ta1);
asstablet2.setText(ta2);
asstablet3.setText(ta3);
asstablet4.setText(ta4);
```

```
dashboard.add(asstablet1);
dashboard.add(asstablet2);
dashboard.add(asstablet3);
dashboard.add(asstablet4);
```

```
conn.close();
```

```

        }
        catch(Exception e)
        {
            System.out.println("    "+e);
        }
        Icon iconclose = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img57.png");
        JButton close= new JButton(iconclose);
        close.setBorderPainted(false);
        close.setFocusPainted(false);
        close.setContentAreaFilled(false);
        close.setBounds(670,500,250,100);

        close.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {

                frame.setVisible(false);
                table d=new table();
                d.tables(userid);

            }
        });

        dashboard.add(close);

        dashboard.add(labeldashass);
        dashboard.add(labeldashapp);
        dashboard.add(labelprofile);

        dashboard.updateUI();
        dashboard.setVisible(true);
        schedule.setVisible(false);
        cancel.setVisible(false);
        order.setVisible(false);
        vieworder.setVisible(false);
    }
});

```

```

b1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        b.setIcon(icon4);
        b1.setIcon(icon5);
        b2.setIcon(icon2);
        b3.setIcon(icon3);
        b4.setIcon(icon8);

        order.removeAll();
        order.revalidate();
        order.repaint();
        order.setVisible(true);

        JPanel item1=new JPanel();
        item1.setBackground(new Color(255, 222, 89));
        item1.setLayout(null);
        item1.setBounds(450,150,450,50);
        JCheckBox c1=new JCheckBox();
        c1.setBackground(new Color(255, 222, 89));
        c1.setForeground(Color.white);
        JLabel itemt1id=new JLabel("-");
        JLabel itemt1name=new JLabel("-");
        JLabel itemt1quantity=new JLabel("-");
        JLabel itemt1price=new JLabel("-");
        JLabel itemt1total=new JLabel("-");
        c1.setBounds(0,0,50,50);
        itemt1id.setBounds(50,0,300,50);
        itemt1name.setBounds(90,0,300,50);
        itemt1quantity.setBounds(180,0,300,50);
        itemt1price.setBounds(270,0,300,50);
        itemt1total.setBounds(360,0,300,50);
        item1.add(c1);
        item1.add(itemt1id);
        item1.add(itemt1name);
        item1.add(itemt1quantity);
        item1.add(itemt1price);
        item1.add(itemt1total);
        order.add(item1);
    }
});

```

```
JPanel item2=new JPanel();
item2.setBackground(new Color(255, 222, 89));
item2.setLayout(null);
item2.setBounds(450,200,450,50);
JCheckBox c2=new JCheckBox();
c2.setBackground(new Color(255, 222, 89));
c2.setForeground(Color.white);
JLabel itemt2id=new JLabel("-");
JLabel itemt2name=new JLabel("-");
JLabel itemt2quantity=new JLabel("-");
JLabel itemt2price=new JLabel("-");
JLabel itemt2total=new JLabel("-");
c2.setBounds(0,0,50,50);
itemt2id.setBounds(50,0,300,50);
itemt2name.setBounds(90,0,300,50);
itemt2quantity.setBounds(180,0,300,50);
itemt2price.setBounds(270,0,300,50);
itemt2total.setBounds(360,0,300,50);
item2.add(c2);
item2.add(itemt2id);
item2.add(itemt2name);
item2.add(itemt2quantity);
item2.add(itemt2price);
item2.add(itemt2total);
order.add(item2);
```

```
JPanel item3=new JPanel();
item3.setBackground(new Color(255, 222, 89));
item3.setLayout(null);
item3.setBounds(450,250,450,50);
JCheckBox c3=new JCheckBox();
c3.setBackground(new Color(255, 222, 89));
c3.setForeground(Color.white);
JLabel itemt3id=new JLabel("-");
JLabel itemt3name=new JLabel("-");
JLabel itemt3quantity=new JLabel("-");
JLabel itemt3price=new JLabel("-");
JLabel itemt3total=new JLabel("-");
c3.setBounds(0,0,50,50);
itemt3id.setBounds(50,0,300,50);
itemt3name.setBounds(90,0,300,50);
```

```
itemt3quantity.setBounds(180,0,300,50);
itemt3price.setBounds(270,0,300,50);
itemt3total.setBounds(360,0,300,50);
item3.add(c3);
item3.add(itemt3id);
item3.add(itemt3name);
item3.add(itemt3quantity);
item3.add(itemt3price);
item3.add(itemt3total);
order.add(item3);
```

```
JPanel item4=new JPanel();
item4.setBackground(new Color(255, 222, 89));
item4.setLayout(null);
item4.setBounds(450,300,450,50);
JCheckBox c4=new JCheckBox();
c4.setBackground(new Color(255, 222, 89));
c4.setForeground(Color.white);
JLabel itemt4id=new JLabel("-");
JLabel itemt4name=new JLabel("-");
JLabel itemt4quantity=new JLabel("-");
JLabel itemt4price=new JLabel("-");
JLabel itemt4total=new JLabel("-");
c4.setBounds(0,0,50,50);
itemt4id.setBounds(50,0,300,50);
itemt4name.setBounds(90,0,300,50);
itemt4quantity.setBounds(180,0,300,50);
itemt4price.setBounds(270,0,300,50);
itemt4total.setBounds(360,0,300,50);
item4.add(c4);
item4.add(itemt4id);
item4.add(itemt4name);
item4.add(itemt4quantity);
item4.add(itemt4price);
item4.add(itemt4total);
order.add(item4);
```

```
JPanel item5=new JPanel();
item5.setBackground(new Color(255, 222, 89));
item5.setLayout(null);
```

```
item5.setBounds(450,350,450,50);
JCheckBox c5=new JCheckBox();
c5.setBackground(new Color(255, 222, 89));
c5.setForeground(Color.white);
JLabel itemt5id=new JLabel("-");
JLabel itemt5name=new JLabel("-");
JLabel itemt5quantity=new JLabel("-");
JLabel itemt5price=new JLabel("-");
JLabel itemt5total=new JLabel("-");
c5.setBounds(0,0,50,50);
itemt5id.setBounds(50,0,300,50);
itemt5name.setBounds(90,0,300,50);
itemt5quantity.setBounds(180,0,300,50);
itemt5price.setBounds(270,0,300,50);
itemt5total.setBounds(360,0,300,50);
item5.add(c5);
item5.add(itemt5id);
item5.add(itemt5name);
item5.add(itemt5quantity);
item5.add(itemt5price);
item5.add(itemt5total);
order.add(item5);

JPanel item6=new JPanel();
item6.setBackground(new Color(255, 222, 89));
item6.setLayout(null);
item6.setBounds(450,400,450,50);

JLabel itemt6total=new JLabel("0");
JLabel itemt6price=new JLabel("Total:");
itemt6price.setBounds(270,0,300,50);
itemt6total.setBounds(360,0,300,50);
item6.add(itemt6price);
item6.add(itemt6total);
order.add(item6);

item1.setVisible(false);
item2.setVisible(false);
```

```
item3.setVisible(false);
item4.setVisible(false);
item5.setVisible(false);
item6.setVisible(false);
```

```
ImageIcon iconord = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img37.png");
JLabel labelord = new JLabel(iconord);
labelord.setBounds(-150,0,700,500);
```

```
ArrayList<String> productlist=new ArrayList<String>();
ArrayList<String> productstock=new ArrayList<String>();
ArrayList<String> productprize=new ArrayList<String>();
ArrayList<String> productid=new ArrayList<String>();
productlist.add("-");
productstock.add("0");
productprize.add("0");
productid.add("0");
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
    Statement st=conn.createStatement();
    String sql = "SELECT * FROM items";
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()){
        productid.add(rs.getString("productid"));
        productlist.add(rs.getString("productname"));
        productstock.add(rs.getString("stock"));
        productprize.add(rs.getString("price"));
    }
    conn.close();
}
catch(Exception e){
    System.out.println(" "+e);
}
String[] productarr=new String[productlist.size()];
productlist.toArray(productarr);
JComboBox product=new JComboBox(productarr);
product.setBounds(190,120,120,30);
```

```
product.setBackground(Color.white);
```

```
JLabel orderprize=new JLabel();  
JLabel orderavailable=new JLabel();  
orderprize.setBounds(190,190,120,30);  
orderavailable.setBounds(190,240,120,30);  
orderprize.setText("-");  
orderavailable.setText("-");
```

```
JTextField quan=new JTextField();  
quan.setBounds(210,315,50,20);  
quan.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
quan.setBackground(new Color(217,217,217));
```

```
Icon iconaddtocart = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img40.png");  
JButton btnaddtocart= new JButton(iconaddtocart);  
btnaddtocart.setBorderPainted(false);  
btnaddtocart.setFocusPainted(false);  
btnaddtocart.setContentAreaFilled(false);  
btnaddtocart.setBounds(100,350,250,100);
```

```
Icon iconremovetocart = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img41.png");  
JButton btnremovetocart= new JButton(iconremovetocart);  
btnremovetocart.setBorderPainted(false);  
btnremovetocart.setFocusPainted(false);  
btnremovetocart.setContentAreaFilled(false);  
btnremovetocart.setBounds(450,500,250,100);
```

```
Icon iconconfirmtocart = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img42.png");  
JButton btnconfirmtocart= new JButton(iconconfirmtocart);  
btnconfirmtocart.setBorderPainted(false);  
btnconfirmtocart.setFocusPainted(false);  
btnconfirmtocart.setContentAreaFilled(false);  
btnconfirmtocart.setBounds(650,500,250,100);
```

```
JPanel cart=new JPanel();  
cart.setBackground(Color.white);  
cart.setLayout(null);
```



```
ImageIcon carts = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img38.png");  
JLabel cartlb = new JLabel(carts);  
cartlb.setBounds(0,0,700,500);  
cartlb.setLayout(null);  
cart.setBounds(320,0,800,800);  
cart.add(cartlb);
```

```
product.addItemListener(new ItemListener() {  
    @Override  
    public void itemStateChanged(ItemEvent ie) {  
        //To change body of generated methods, choose Tools | Templates.  
        String  
prize=productprize.get(productlist.indexOf(product.getSelectedItem().toString()));  
        String  
available=productstock.get(productlist.indexOf(product.getSelectedItem().toString()  
));  
        orderprize.setText(prize);  
        orderavailable.setText(available);  
    }  
});
```

```
btnaddtocart.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        System.out.println("working");  
        String productname=product.getSelectedItem().toString();  
        String price=orderprize.getText().toString();  
        String available=orderavailable.getText().toString();  
        String quantity=quan.getText().toString();  
        String  
total=String.valueOf(Integer.parseInt(quantity)*Integer.parseInt(price));
```

```

        int totalnum=Integer.parseInt(quantity)*Integer.parseInt(price);
        int totalvalue=Integer.parseInt(itemt6total.getText()+totalnum);
        String totalvaluestring=String.valueOf(totalvalue);
        if(productname.equals('-')){
            Error e=new Error();
            e.errordis("Choose product","Check again");
        }
        else
        if(productname.equals(itemt1name.getText().toString())||productname.equals(itemt
        2name.getText().toString())||productname.equals(itemt3name.getText().toString())|
        |productname.equals(itemt4name.getText().toString())||productname.equals(itemt
        5name.getText().toString())){
            Error e=new Error();
            e.errordis("Sorry product is entered","again");

        }
        else if(totalitemscart>=5){
            Error e=new Error();
            e.errordis("Cart is full","Cant add items");
        }
        else if((Integer.parseInt(quantity)>Integer.parseInt(available) )||
        (Integer.parseInt(quantity)==0)){
            Error e=new Error();
            e.errordis("Stock not available","Can't add items");
        }
        else{
            if(totalitemscart==0){

                itemt1id.setText("1");
                itemt1name.setText(productname);
                itemt1quantity.setText(quantity);
                itemt1price.setText(price);
                itemt1total.setText(total);

                item1.setVisible(true);

                itemt6total.setText(totalvaluestring);
                item6.setVisible(true);

                System.out.println("cart1");
                order.repaint();
            }
        }
    }
}

```

```

        order.updateUI();
        totalitemscart=totalitemscart+1;
    }
    else if(totalitemscart==1){
        itemt2id.setText("2");
        itemt2name.setText(productname);
        itemt2quantity.setText(quantity);
        itemt2price.setText(price);
        itemt2total.setText(total);

        item1.setVisible(true);
        item2.setVisible(true);

        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);

        System.out.println("cart1");
        order.repaint();
        order.updateUI();
        totalitemscart=totalitemscart+1;
    }
    else if(totalitemscart==2){
        itemt3id.setText("3");
        itemt3name.setText(productname);
        itemt3quantity.setText(price);
        itemt3price.setText(quantity);
        itemt3total.setText(total);

        item1.setVisible(true);
        item2.setVisible(true);
        item3.setVisible(true);

        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);

        System.out.println("cart1");
        order.repaint();
        order.updateUI();
        totalitemscart=totalitemscart+1;
    }
    else if(totalitemscart==3){

```

```
        itemt4id.setText("4");
        itemt4name.setText(productname);
        itemt4quantity.setText(quantity);
        itemt4price.setText(price);
        itemt4total.setText(total);

        item1.setVisible(true);
        item2.setVisible(true);
        item3.setVisible(true);
        item4.setVisible(true);

        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);

        System.out.println("cart1");
        order.repaint();
        order.updateUI();
        totalitemscart=totalitemscart+1;
    }
    else if(totalitemscart==4){
        itemt5id.setText("5");
        itemt5name.setText(productname);
        itemt5quantity.setText(quantity);
        itemt5price.setText(price);
        itemt5total.setText(total);

        item1.setVisible(true);
        item2.setVisible(true);
        item3.setVisible(true);
        item4.setVisible(true);
        item5.setVisible(true);

        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);

        System.out.println("cart1");
        order.repaint();
        order.updateUI();
        totalitemscart=totalitemscart+1;
    }
```

```
}
```

```
}
```

```
});
```

```
btnremovetocart.addActionListener(new ActionListener() {
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent ae) {
```

```
        if(c1.isSelected()){
```

```
            itemt1name.setText(itemt2name.getText());
```

```
            itemt1quantity.setText(itemt2quantity.getText());
```

```
            itemt1price.setText(itemt2price.getText());
```

```
            itemt1total.setText(itemt2total.getText());
```

```
            itemt2name.setText(itemt3name.getText());
```

```
            itemt2quantity.setText(itemt3quantity.getText());
```

```
            itemt2price.setText(itemt3price.getText());
```

```
            itemt2total.setText(itemt3total.getText());
```

```
            itemt3name.setText(itemt4name.getText());
```

```
            itemt3quantity.setText(itemt4quantity.getText());
```

```
            itemt3price.setText(itemt4price.getText());
```

```
            itemt3total.setText(itemt4total.getText());
```

```
            itemt4name.setText(itemt5name.getText());
```

```
            itemt4quantity.setText(itemt5quantity.getText());
```

```
            itemt4price.setText(itemt5price.getText());
```

```
            itemt4total.setText(itemt5total.getText());
```

```
            totalitemscart=totalitemscart-1;
```

```
        }
```

```
        if(c2.isSelected()){
```

```
            itemt2name.setText(itemt3name.getText());
```

```
            itemt2quantity.setText(itemt3quantity.getText());
```

```

        itemt2price.setText(itemt3price.getText());
        itemt2total.setText(itemt3total.getText());

        itemt3name.setText(itemt4name.getText());
        itemt3quantity.setText(itemt4quantity.getText());
        itemt3price.setText(itemt4price.getText());
        itemt3total.setText(itemt4total.getText());

        itemt4name.setText(itemt5name.getText());
        itemt4quantity.setText(itemt5quantity.getText());
        itemt4price.setText(itemt5price.getText());
        itemt4total.setText(itemt5total.getText());

        totalitemscart=totalitemscart-1;

    }
    if(c3.isSelected()){

        itemt3name.setText(itemt4name.getText());
        itemt3quantity.setText(itemt4quantity.getText());
        itemt3price.setText(itemt4price.getText());
        itemt3total.setText(itemt4total.getText());

        itemt4name.setText(itemt5name.getText());
        itemt4quantity.setText(itemt5quantity.getText());
        itemt4price.setText(itemt5price.getText());
        itemt4total.setText(itemt5total.getText());

        totalitemscart=totalitemscart-1;

    }
    if(c4.isSelected()){

        itemt4name.setText(itemt5name.getText());
        itemt4quantity.setText(itemt5quantity.getText());
        itemt4price.setText(itemt5price.getText());
        itemt4total.setText(itemt5total.getText());

        totalitemscart=totalitemscart-1;

    }

```

```

        if(c5.isSelected()){
            totalitemscart=totalitemscart-1;

        }

        if(totalitemscart==0){
            item1.setVisible(false);
            item2.setVisible(false);
            item3.setVisible(false);
            item4.setVisible(false);
            item5.setVisible(false);

            item6.setVisible(false);
        }
        else if(totalitemscart==1){
            item1.setVisible(true);
            item2.setVisible(false);
            item3.setVisible(false);
            item4.setVisible(false);
            item5.setVisible(false);

            int totalvaluenum=Integer.parseInt(itemt1total.getText());
            String totalvaluestring=String.valueOf(totalvaluenum);
            itemt6total.setText(totalvaluestring);
            item6.setVisible(true);
        }
        else if(totalitemscart==2){
            item1.setVisible(true);
            item2.setVisible(true);
            item3.setVisible(false);
            item4.setVisible(false);
            item5.setVisible(false);

            int
totalvaluenum=Integer.parseInt(itemt1total.getText()+Integer.parseInt(itemt2total.
getText()));
            String totalvaluestring=String.valueOf(totalvaluenum);
            itemt6total.setText(totalvaluestring);
            item6.setVisible(true);

```

```

    }
    else if(totalitemscart==3){
        item1.setVisible(true);
        item2.setVisible(true);
        item3.setVisible(true);
        item4.setVisible(false);
        item5.setVisible(false);

        int
totalvaluenum=Integer.parseInt(itemt1total.getText()+Integer.parseInt(itemt2total.
getText()+Integer.parseInt(itemt3total.getText());
        String totalvaluestring=String.valueOf(totalvaluenum);
        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);
    }
    else if(totalitemscart==4){
        item1.setVisible(true);
        item2.setVisible(true);
        item3.setVisible(true);
        item4.setVisible(true);
        item5.setVisible(false);

        int
totalvaluenum=Integer.parseInt(itemt1total.getText()+Integer.parseInt(itemt2total.
getText()+Integer.parseInt(itemt3total.getText()+Integer.parseInt(itemt4total.getTe
xt());

        String totalvaluestring=String.valueOf(totalvaluenum);
        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);
    }
    else if(totalitemscart==5){
        item1.setVisible(true);
        item2.setVisible(true);
        item3.setVisible(true);
        item4.setVisible(true);
        item5.setVisible(true);

        int
totalvaluenum=Integer.parseInt(itemt1total.getText()+Integer.parseInt(itemt2total.
getText()+Integer.parseInt(itemt3total.getText()+Integer.parseInt(itemt4total.getTe
xt()+Integer.parseInt(itemt4total.getText());

```



```

        String totalvaluestring=String.valueOf(totalvaluenum);
        itemt6total.setText(totalvaluestring);
        item6.setVisible(true);
    }

}

});

btnconfirmcart.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        if(totalitemscart==5){
            try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

                Statement st=conn.createStatement();
                Timestamp instant=Timestamp.from(Instant.now());
                String delid="DI "+userid+instant;

                String s1="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','1','"+itemt1name.getText().toString()+"','"+itemt1quantity.getText
().toString()+"','"+itemt1price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt1name.getText().toString()))+"');
";

                st.executeUpdate(s1);
                String s2="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','2','"+itemt2name.getText().toString()+"','"+itemt2quantity.getText
().toString()+"','"+itemt2price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt2name.getText().toString()))+"');
";

                st.executeUpdate(s2);
                String s3="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','3','"+itemt3name.getText().toString()+"','"+itemt3quantity.getText

```

```

().toString()+"', '"+itemt3price.getText().toString()+"', '"+userid+"', 'not
delivered', '"+productid.get(productlist.indexOf(itemt3name.getText().toString()))+''";
";

        st.executeUpdate(s3);
        String s4="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"', '4', '"+itemt4name.getText().toString()+"', '"+itemt4quantity.getText
().toString()+"', '"+itemt4price.getText().toString()+"', '"+userid+"', 'not
delivered', '"+productid.get(productlist.indexOf(itemt4name.getText().toString()))+''";
";

        st.executeUpdate(s4);
        String s5="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"', '5', '"+itemt5name.getText().toString()+"', '"+itemt5quantity.getText
().toString()+"', '"+itemt5price.getText().toString()+"', '"+userid+"', 'not
delivered', '"+productid.get(productlist.indexOf(itemt5name.getText().toString()))+''";
";

        st.executeUpdate(s5);
        System.out.println(itemt1quantity.getText().toString());
        System.out.println(itemt2quantity.getText().toString());
        System.out.println(itemt3quantity.getText().toString());
        System.out.println(itemt4quantity.getText().toString());
        System.out.println(itemt5quantity.getText().toString());

        String q1="UPDATE items set stock=(stock-
"+itemt1quantity.getText().toString()+") where
productname='"+itemt1name.getText().toString()+"''";
        st.executeUpdate(q1);
        String q2="UPDATE items set stock=(stock-
"+itemt2quantity.getText().toString()+") where
productname='"+itemt2name.getText().toString()+"''";
        st.executeUpdate(q2);
        String q3="UPDATE items set stock=(stock-
"+itemt3quantity.getText().toString()+") where
productname='"+itemt3name.getText().toString()+"''";
        st.executeUpdate(q3);
        String q4="UPDATE items set stock=(stock-
"+itemt4quantity.getText().toString()+") where
productname='"+itemt4name.getText().toString()+"''";
        st.executeUpdate(q4);

```

```

String q5="UPDATE items set stock=(stock-
"+itemt5quantity.getText().toString()+") where
productname='"+itemt5name.getText().toString()+"';
st.executeUpdate(q5);

```

```

messageboxplace m=new messageboxplace();
m.message();

```

```

System.out.println("Success!");
conn.close();
}
catch(Exception e){
System.out.println(" "+e);
}

```

```

}
else if(totalitemscart==4){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

Statement st=conn.createStatement();
Timestamp instant=Timestamp.from(Instant.now());
String delid="DI "+userid+instant;
String s1="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','1','"+itemt1name.getText().toString()+"','"+itemt1quantity.getText
().toString()+"','"+itemt1price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt1name.getText().toString()))+"');
";

st.executeUpdate(s1);
String s2="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','2','"+itemt2name.getText().toString()+"','"+itemt2quantity.getText
().toString()+"','"+itemt2price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt2name.getText().toString()))+"');
";

```

```

        st.executeUpdate(s2);
        String s3="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','3','"+itemt3name.getText().toString()+"','"+itemt3quantity.getText
().toString()+"','"+itemt3price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt3name.getText().toString()))+"');
";

        st.executeUpdate(s3);
        String s4="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','4','"+itemt4name.getText().toString()+"','"+itemt4quantity.getText
().toString()+"','"+itemt4price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt4name.getText().toString()))+"');
";

        st.executeUpdate(s4);

        System.out.println(itemt1quantity.getText().toString());
        System.out.println(itemt2quantity.getText().toString());
        System.out.println(itemt3quantity.getText().toString());
        System.out.println(itemt4quantity.getText().toString());

```

```

        String q1="UPDATE items set stock=(stock-
"+itemt1quantity.getText().toString()+") where
productname='"+itemt1name.getText().toString()+"'";
        st.executeUpdate(q1);
        String q2="UPDATE items set stock=(stock-
"+itemt2quantity.getText().toString()+") where
productname='"+itemt2name.getText().toString()+"'";
        st.executeUpdate(q2);
        String q3="UPDATE items set stock=(stock-
"+itemt3quantity.getText().toString()+") where
productname='"+itemt3name.getText().toString()+"'";
        st.executeUpdate(q3);
        String q4="UPDATE items set stock=(stock-
"+itemt4quantity.getText().toString()+") where
productname='"+itemt4name.getText().toString()+"'";
        st.executeUpdate(q4);

```

```

        messageboxplace m=new messageboxplace();
        m.message();

        System.out.println("Success!");
        conn.close();
    }
    catch(Exception e){
        System.out.println("    "+e);
    }
}

else if(totalitemscart==3){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        Timestamp instant=Timestamp.from(Instant.now());
        String delid="DI "+userid+instant;
        String s1="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','1','"+itemt1name.getText().toString()+"','"+itemt1quantity.getText
().toString()+"','"+itemt1price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt1name.getText().toString()))+"');
";

        st.executeUpdate(s1);
        String s2="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','2','"+itemt2name.getText().toString()+"','"+itemt2quantity.getText
().toString()+"','"+itemt2price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt2name.getText().toString()))+"');
";

        st.executeUpdate(s2);
        String s3="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','3','"+itemt3name.getText().toString()+"','"+itemt3quantity.getText

```

```
()).toString()+"','"+itemt3price.getText().toString()+"','"+userid+"','not  
delivered','"+productid.get(productlist.indexOf(itemt3name.getText().toString()))+"");  
";
```

```
st.executeUpdate(s3);
```

```
System.out.println(itemt1quantity.getText().toString());
```

```
System.out.println(itemt2quantity.getText().toString());
```

```
System.out.println(itemt3quantity.getText().toString());
```

```
String q1="UPDATE items set stock=(stock-  
"+itemt1quantity.getText().toString()+") where  
productname='"+itemt1name.getText().toString()+"'";
```

```
st.executeUpdate(q1);
```

```
String q2="UPDATE items set stock=(stock-  
"+itemt2quantity.getText().toString()+") where  
productname='"+itemt2name.getText().toString()+"'";
```

```
st.executeUpdate(q2);
```

```
String q3="UPDATE items set stock=(stock-  
"+itemt3quantity.getText().toString()+") where  
productname='"+itemt3name.getText().toString()+"'";
```

```
st.executeUpdate(q3);
```

```
messageboxplace m=new messageboxplace();  
m.message();
```

```
System.out.println("Success!");
```

```
conn.close();
```

```
}
```

```
catch(Exception e){
```

```
System.out.println(" "+e);
```

```
}
```

```
}
```

```

else if(totalitemscart==2){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        Timestamp instant=Timestamp.from(Instant.now());
        String delid="DI "+userid+instant;
        String s1="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','1','"+itemt1name.getText().toString()+"','"+itemt1quantity.getText
().toString()+"','"+itemt1price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt1name.getText().toString()))+"');
";

        st.executeUpdate(s1);
        String s2="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','2','"+itemt2name.getText().toString()+"','"+itemt2quantity.getText
().toString()+"','"+itemt2price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt2name.getText().toString()))+"');
";

        st.executeUpdate(s2);

        System.out.println(itemt1quantity.getText().toString());
        System.out.println(itemt2quantity.getText().toString());


        String q1="UPDATE items set stock=(stock-
"+itemt1quantity.getText().toString()+") where
productname='"+itemt1name.getText().toString()+"'";
        st.executeUpdate(q1);
        String q2="UPDATE items set stock=(stock-
"+itemt2quantity.getText().toString()+") where
productname='"+itemt2name.getText().toString()+"'";
        st.executeUpdate(q2);

        messageboxplace m=new messageboxplace();

```

```

        m.message();

        System.out.println("Success!");
        conn.close();
    }
    catch(Exception e){
        System.out.println("    "+e);
    }

}
else if(totalitemscart==1){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        Timestamp instant=Timestamp.from(Instant.now());
        String delid="DI "+userid+instant;
        String s1="insert into
ecommerce(orderid,no,productname,quantity,price,userid,status,productid)
values('"+delid+"','1','"+itemt1name.getText().toString()+"','"+itemt1quantity.getText
().toString()+"','"+itemt1price.getText().toString()+"','"+userid+"','not
delivered','"+productid.get(productlist.indexOf(itemt1name.getText().toString()))+"');
";

        st.executeUpdate(s1);

        System.out.println(itemt1quantity.getText().toString());

        String q1="UPDATE items set stock=(stock-
"+itemt1quantity.getText().toString()+") where
productname='"+itemt1name.getText().toString()+"'";
        st.executeUpdate(q1);

        messageboxplace m=new messageboxplace();
        m.message();

```



```

        System.out.println("Success!");
        conn.close();
    }
    catch(Exception e){
        System.out.println("    "+e);
    }

    }
    b1.doClick();
    totalitemscart=0;

    }
    });

    order.add(btnconfirmtocart);
    order.add(btnremovetocart);
    order.add(btnaddtocart);
    order.add(qua n);
    order.add(cart);
    order.add(orderprize);
    order.add(orderavailable);
    order.add(product);
    order.add(labelord);

        order.updateUI();
        dashboard.setVisible(false);
        schedule.setVisible(false);
        cancel.setVisible(false);
        order.setVisible(true );
    }
    });

    b2.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {

            b.setlcon(icon4);

```

```

        b1.setIcon(icon1);
        b2.setIcon(icon6);
        b3.setIcon(icon3);
        b4.setIcon(icon8);

        schedule.removeAll();
        schedule.revalidate();
        schedule.repaint();
        schedule.setVisible(true);

        Icon iconapp = new
        ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img23.png");
        JButton bapp= new JButton(iconapp);
        bapp.setBorderPainted(false);
        bapp.setFocusPainted(false);
        bapp.setContentAreaFilled(false);
        bapp.setBounds(550,500,250,100);

        ImageIcon iconapplb = new
        ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img21.png");
        JLabel labelapp = new JLabel(iconapplb);
        labelapp.setBounds(360,10,650,500);

        JTextField tfappveh=new JTextField();
        tfappveh.setBounds(570,140,120,30);
        tfappveh.setBorder(javax.swing.BorderFactory.createEmptyBorder());
        tfappveh.setBackground(new Color(217,217,217));

        ArrayList<String> workshoplist=new ArrayList<String>();
        workshoplist.add("_");
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
            03");
            Statement st=conn.createStatement();
            String sql = "SELECT name FROM admin";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next()){
                workshoplist.add(rs.getString("name"));
            }
        }
    }

```

```

    }
    System.out.println(workshoplist);

    conn.close();
}
catch(Exception e)
{
    System.out.println(" "+e);
}

String[] workshoparr=new String[workshoplist.size()];
workshoplist.toArray(workshoparr);

JComboBox tfappwor=new JComboBox(workshoparr);
tfappwor.setBounds(530,240,160,30);
tfappwor.setBorder(javax.swing.BorderFactory.createEmptyBorder());
tfappwor.setBackground(Color.white);

JPanel panelser=new JPanel();
panelser.setBounds(570,320,120,30);
panelser.setLayout(null);
panelser.setBackground(Color.white);

JPanel paneldate=new JPanel();
paneldate.setBounds(850,300,120,30);
paneldate.setLayout(null);
paneldate.setBackground(Color.white);

JPanel paneltime=new JPanel();
paneltime.setBounds(850,370,120,30);
paneltime.setLayout(null);
paneltime.setBackground(Color.white);

tfappwor.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools | Templates.
        Component[] components=panelser.getComponents();
        for(Component component :components){

            panelser.remove(component);

```

```

    }
    panelser.revalidate();
    panelser.repaint();
    panelser.setVisible(false);
    int adminid=workshoplist.indexOf(tfappwor.getSelectedItem().toString());
    ArrayList<String> servicelist=new ArrayList<String>();
    servicelist.add("-");
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String sql = "SELECT service from workshop where
adminid='"+adminid+"'";
        ResultSet rs  = st.executeQuery(sql);
        while(rs.next()){
            servicelist.add(rs.getString("service"));
        }
        System.out.println("sucess district");
        System.out.println(servicelist);
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
    String[] servicearr=new String[servicelist.size()];
    servicelist.toArray(servicearr);
    JComboBox service=new JComboBox(servicearr);

    service.setBounds(0,0,120,30);
    service.setBackground(Color.white);

    service.addItemListener(new ItemListener() {
        @Override
        public void itemStateChanged(ItemEvent ie) {
            //To change body of generated methods, choose Tools |
Templates.
            serviceinput=service.getSelectedItem().toString();
            System.out.println(serviceinput);

```

```

        }
    });

    panelser.setBackground(Color.white);
    panelser.add(service);

    Component[] componentsd=panelser.getComponents();
    for(Component component :componentsd){

        paneldate.remove(component);
    }
    paneldate.revalidate();
    paneldate.repaint();
    paneldate.setVisible(false);

    ArrayList<String> datelist=new ArrayList<String>();
    datelist.add("-");
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String sql = "SELECT DISTINCT date from schedule where
adminid='"+adminid+"';";
        ResultSet rs  = st.executeQuery(sql);
        while(rs.next()){
            datelist.add(rs.getString("date"));
        }
        System.out.println("sucess date");
        System.out.println(datelist);
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
    String[] datearr=new String[datelist.size()];
    datelist.toArray(datearr);
    JComboBox date=new JComboBox(datearr);

```

```

date.setBounds(0,0,120,30);
date.setBackground(Color.white);

```

```

date.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools |

```

Templates.

```

dateinput=date.getSelectedItem().toString();
System.out.println(serviceinput);
Component[] components=paneltime.getComponents();
for(Component component :components){
    paneltime.remove(component);
}
paneltime.revalidate();
paneltime.repaint();
paneltime.setVisible(false);

```

```

ArrayList<String> timelist=new ArrayList<String>();
timelist.add("-");
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn =

```

```

DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

```

```

        Statement st=conn.createStatement();
        int
adminid=workshoplist.indexOf(tfappwor.getSelectedItem().toString());
        String sql = "SELECT DISTINCT time from schedule where
adminid='"+adminid+"' and date='"+date.getSelectedItem().toString()+"'";
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            timelist.add(rs.getString("time"));
        }
        System.out.println("sucess time");
        System.out.println(timelist);
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
}

```

Templates.

```
String[] timearr=new String[timelist.size()];
timelist.toArray(timearr);
JComboBox time=new JComboBox(timearr);
time.setBounds(0,0,120,30);
time.setBackground(Color.white);
time.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools |
        timeinput=time.getSelectedItem().toString();
        System.out.println(timeinput);
    }
});

paneltime.setBackground(Color.white);
paneltime.add(time);
paneltime.setVisible(true);

schedule.updateUI();
panelser.setVisible(true);
panelser.updateUI();
paneldate.updateUI();
paneltime.updateUI();

});
paneldate.setBackground(Color.white);
paneldate.add(date);
paneldate.setVisible(true);

schedule.updateUI();
panelser.setVisible(true);
panelser.updateUI();
paneldate.updateUI();
```

```

    }
});

ArrayList<String> vehiclebrandlist=new ArrayList<String>();
vehiclebrandlist.add("-");
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

    Statement st=conn.createStatement();
    String sql = "SELECT distinct vehiclebrand from vehicle;";
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()){
        vehiclebrandlist.add(rs.getString("vehiclebrand"));
    }
    conn.close();
}
catch(Exception e){
    System.out.println(" "+e);
}
String[] vehiclebrandarr=new String[vehiclebrandlist.size()];
vehiclebrandlist.toArray(vehiclebrandarr);
JComboBox vehiclebrand=new JComboBox(vehiclebrandarr);
vehiclebrand.setBounds(840,140,120,30);
vehiclebrand.setBackground(Color.white);

JPanel panelvehiclemodel=new JPanel();
panelvehiclemodel.setBounds(840,220,120,30);
panelvehiclemodel.setLayout(null);
panelvehiclemodel.setBackground(Color.white);

vehiclebrand.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools | Templates.
        Component[] components=panelvehiclemodel.getComponents();
        for(Component component :components){

            panelvehiclemodel.remove(component);
        }
    }
});

```



```

        panelvehiclemodel.revalidate();
        panelvehiclemodel.repaint();
        String vehiclebr=vehiclebrand.getSelectedItem().toString();
        panelvehiclemodel.setVisible(false);
        ArrayList<String> vehmlist=new ArrayList<String>();
        vehmlist.add("-");
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st=conn.createStatement();
            String sql = "SELECT vehiclemodel from vehicle where
vehiclebrand='"+vehiclebr+"'";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next()){
                vehmlist.add(rs.getString("vehiclemodel"));
            }
            System.out.println("sucess district");
            System.out.println(vehmlist);
            conn.close();
        }
        catch(Exception e){
            System.out.println(" "+e);
        }
        String[] vehmarr=new String[vehmlist.size()];
        vehmlist.toArray(vehmarr);
        JComboBox vehmodel=new JComboBox(vehmarr);
        vehmodel.setBounds(0,0,120,30);
        vehmodel.setBackground(Color.white);

        vehmodel.addItemListener(new ItemListener() {
            @Override
            public void itemStateChanged(ItemEvent ie) {
                //To change body of generated methods, choose Tools |
                vehmodelinput=vehmodel.getSelectedItem().toString();
                System.out.println(vehmodelinput);
            }
        });

```

Templates.

```

        panelvehicmodel.setBackground(Color.white);
        panelvehicmodel.add(vehmodel);
        panelvehicmodel.updateUI();
        panelvehicmodel.setVisible(true);
        schedule.updateUI();

    }
});

```

```

bapp.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        String vehno=tfappveh.getText();
        String workshop=tfappwor.getSelectedltem().toString();
        String vehbrand=vehiclebrand.getSelectedltem().toString();
        Boolean error=false;
        if(vehno.isEmpty()){
            Error e=new Error();
            e.errordis("Fill Vehicle no","Check again");
            error=true;

        }
        else if(vehbrand.isEmpty() || vehbrand.equals('-')){
            Error e=new Error();
            e.errordis("Select Vehicle brand","Check again");
            error=true;

        }
        else if(workshop.isEmpty() || workshop.equals('-')){
            Error e=new Error();
            e.errordis("Select workshop","Check again");
            error=true;

        }
        else if(serviceinput.isEmpty() || serviceinput.equals('-')){
            Error e=new Error();
            e.errordis("Select service need from workshop","Check again");
            error=true;

        }
    }
}

```

```

else if(dateinput.isEmpty() || dateinput.equals('-')){
    Error e=new Error();
    e.errordis("Select date","Check again");
    error=true;

}
else if(timeinput.isEmpty() || timeinput.equals('-')){
    Error e=new Error();
    e.errordis("Select time","Check again");
    error=true;

}
else if(vehmodelinput.isEmpty() || vehmodelinput.equals('-')){
    Error e=new Error();
    e.errordis("Select vehicle model","Check again");
    error=true;

}
else{
    boolean empty=true;
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String id=timeinput+" "+dateinput+"
"+tfappwor.getSelectedItemAt().toString();
        String sql = "SELECT id from appointments where id='"+id+"'";
        ResultSet rs = st.executeQuery(sql);
        while(rs.next()){
            empty=false;
            System.out.println("alreadyexist");
        }
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
    if(!empty){
        Error e=new Error();

```

```

        e.errordis("That time slot is already booked", "Pls choose another
option");
        error=true;
    }
    else{
        boolean empty1=true;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st=conn.createStatement();

            String sql = "select userid,date from appointments where
userid='"+userid+"' and date='"+dateinput+"'";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next()){
                empty1=false;
                System.out.println("alreadyexist");
            }
            conn.close();
        }
        catch(Exception e){
            System.out.println(" "+e);
        }
        if(!empty1){
            Error e=new Error();
            e.errordis("Already an appointment is booked", "for the particular
day");
            error=true;
        }

    }

}

if(error==false){
    try{
        Class.forName("com.mysql.jdbc.Driver");

```

```

        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        int
adminid=workshoplist.indexOf(tfappwor.getSelectedItem().toString());
        String id=timeinput+ " "+dateinput+"
"+tfappwor.getSelectedItem().toString();
        String s="INSERT INTO
appointments(userid,vehicleno,adminid,service,status,id,vehiclebrand,vehicmodel,
date)
VALUES('"+userid+"','"+vehno+"','"+a dminid+"','"+serviceinput+"','Approved','"+id+"',
 '"+vehbrand+"','"+vehmodelinput+"','"+dateinput+"');";
        st.executeUpdate(s);
        System.out.println("Success!");
        messageboxapp m=new messageboxapp();
        m.message();
        conn.close();
    }
    catch(Exception e){
        System.out.println("    "+e);
    }

}
b2.doClick();
}

});

```

```

        ImageIcon iconas = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img20.png");
        JLabel labelas = new JLabel(iconas);
        labelas.setBounds(-50,10,400,500);

```

```

        Icon iconres = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img22.png");
        JButton bres= new JButton(iconres);
        bres.setBorderPainted(false);
        bres.setFocusPainted(false);
        bres.setContentAreaFilled(false);

```

```

        bres.setBounds(50,500,250,100);

        ArrayList<String> servicelist=new ArrayList<String>();
        servicelist.add("-");
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject", "root", "Ram
03");
            Statement st=conn.createStatement();
            String sql = "select distinct service from workshop";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next()){
                servicelist.add(rs.getString("service"));
            }
            System.out.println(servicelist);

            conn.close();
        }
        catch(Exception e)
        {
            System.out.println(" "+e);
        }

        String[] servicearr=new String[servicelist.size()];
        servicelist.toArray(servicearr);

        JComboBox tfassser=new JComboBox(servicearr);
        tfassser.setBounds(145,130,180,30);
        tfassser.setBorder(javax.swing.BorderFactory.createEmptyBorder());
        tfassser.setBackground(Color.white);

        JComboBox vehiclebrandass=new JComboBox(vehiclebrandarr);
        vehiclebrandass.setBounds(30,420,120,30);
        vehiclebrandass.setBackground(Color.white);

        JPanel panelvehicmodelass=new JPanel();
        panelvehicmodelass.setBounds(190,420,120,30);
        panelvehicmodelass.setLayout(null);
        panelvehicmodelass.setBackground(Color.white);

```

```

vehiclebrandass.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools | Templates.
        Component[] components=panelvehicmodelass.getComponents();
        for(Component component :components){

            panelvehicmodelass.remove(component);
        }
        panelvehicmodelass.revalidate();
        panelvehicmodelass.repaint();
        String vehiclebr=vehiclebrandass.getSelectedItem().toString();
        panelvehicmodelass.setVisible(false);
        ArrayList<String> vehmlist=new ArrayList<String>();
        vehmlist.add("-");
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st=conn.createStatement();
            String sql = "SELECT vehicmodel from vehicle where
vehiclebrand='"+vehiclebr+"'";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next()){
                vehmlist.add(rs.getString("vehicmodel"));
            }
            System.out.println("sucess district");
            System.out.println(vehmlist);
            conn.close();
        }
        catch(Exception e){
            System.out.println(" "+e);
        }
        String[] vehmarr=new String[vehmlist.size()];
        vehmlist.toArray(vehmarr);
        JComboBox vehmodelass=new JComboBox(vehmarr);
        vehmodelass.setBounds(0,0,120,30);
        vehmodelass.setBackground(Color.white);

```

```

        vehmodelass.addItemListener(new ItemListener() {
            @Override
            public void itemStateChanged(ItemEvent ie) {
                //To change body of generated methods, choose Tools |
                Templates.
                vehmodelinputass=vehmodelass.getSelectedItem().toString();
                System.out.println(vehmodelinputass);
            }
        });

        panelvehicmodelass.setBackground(Color.white);
        panelvehicmodelass.add(vehmodelass);
        panelvehicmodelass.updateUI();
        panelvehicmodelass.setVisible(true);
        schedule.updateUI();

    }
});

JTextField tfassloc=new JTextField();
JTextField tfasspin=new JTextField();

tfassloc.setBounds(40,210,250,40);
tfasspin.setBounds(40,320,100,30);

tfassloc.setBorder(javax.swing.BorderFactory.createEmptyBorder());
tfasspin.setBorder(javax.swing.BorderFactory.createEmptyBorder());

tfassloc.setBackground(new Color(217,217,217));
tfasspin.setBackground(new Color(217,217,217));

bres.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        boolean error=false;
        String address=tfassloc.getText().toString();
        String pincode=tfasspin.getText().toString();
        String ser=(String)tfassser.getSelectedItem();
        String vehb=(String)vehiclebrandass.getSelectedItem();
    }
});

```



```
Pattern p2=Pattern.compile("[^0-9]");
Matcher m2=p2.matcher(pincod);
boolean pinmatch=m2.find();

if(address.isEmpty()){
    Error e=new Error();
    e.erroris("Enter the address","Check again");
    error=true;
}
else if(pincod.isEmpty()){
    Error e=new Error();
    e.erroris("Enter the pincod","Check again");
    error=true;
}
else if(!(pincod.length()==6)){
    Error e=new Error();
    e.erroris("pincod can be of 6 digits","Check again");
    error=true;
}
else if(pinmatch){
    Error e=new Error();
    e.erroris("pincod can be only of numbers","Check again");
    error=true;
}
else if(scr.isEmpty() || scr.equals('-')){
    Error e=new Error();
    e.erroris("Select the service","Check again");
    error=true;
}
else if(vehb.isEmpty() || vehb.equals('-')){
    Error e=new Error();
    e.erroris("Select the vehicle brand","Check again");
    error=true;
}
else if(vehmodelinputass.isEmpty() || vehmodelinputass.equals('-')){
    Error e=new Error();
    e.erroris("Select the vehicle model","Check again");
    error=true;
}
else{
```

```

        boolean notavailable=true;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st=conn.createStatement();

            String sql = "SELECT
workerid,workername,workermobileno,workerstatus,pincode,distance,time,service
from assistpincode ap JOIN workshop w ON ap.adminid=w.adminid JOIN workers wk
ON ap.adminid=wk.adminid where wk.workerstatus='free' and w.service='"+ser+"
and ap.pincode='"+pincode+"';";
            ResultSet rs  = st.executeQuery(sql);
            while(rs.next()){
                notavailable=false;
            }
            conn.close();
        }
        catch(Exception e){
            System.out.println(" "+e);
        }
        if(notavailable){
            Error e=new Error();
            e.errordis("All workers are busy","Try after some time");
            error=true;
        }

```

```

        boolean empty=true;
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st=conn.createStatement();
            String sql = "select userid,status from assist where userid='"+userid+"
and status='Approved'";
            ResultSet rs  = st.executeQuery(sql);
            while(rs.next()){
                empty=false;

```

```

        System.out.println("alreadyexist");
    }
    conn.close();
}
catch(Exception e){
    System.out.println("    "+e);
}
if(!empty){
    Error e=new Error();
    e.erroris("Already response is registered","Pls wait!!!");
    error=true;
}

}

if(error==false){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();

        String sql = "SELECT workerid from assistpincode ap JOIN workshop w
ON ap.adminid=w.adminid JOIN workers wk ON ap.adminid=wk.adminid where
wk.workerstatus='free' and w.service='"+ser+"' and ap.pincod='"+pincod+"'";
        ResultSet rs  = st.executeQuery(sql);
        int workerid=0;
        while(rs.next()){
            workerid=rs.getInt("workerid");
            break;
        }

        String s = "SELECT adminid from workers where
workerid='"+workerid+"'";
        ResultSet r  = st.executeQuery(s);
        int adminid=0;
        while(r.next()){
            adminid=r.getInt("adminid");
            break;

```

```

    }
    Timestamp instant=Timestamp.from(Instant.now());
    String s1="Insert into
assist(timestamp,userid,adminid,workerid,service,landmark,pincode,vehiclebrand,vehic
lemodel,status)
values('"+instant+"','"+userid+"','"+adminid+"','"+workerid+"','"+tfassser.getSelectedI
tem()+"','"+address+"','"+pincode+"','"+vehb+"','"+vehmodelinputass+"','Approved')"
;

    st.executeUpdate(s1);

    String s2="UPDATE workers set workerstatus='not free' where
workerid='"+workerid+"';";
    st.executeUpdate(s2);

    String s3 = "SELECT name from admin where adminid='"+adminid+"';";
    ResultSet r1 = st.executeQuery(s3);
    String adminname="";
    while(r1.next()){
        adminname=r1.getString("name");
    }
    //SELECT * from workers w JOIN assistpincode a ON
w.adminid=a.adminid
    String s4 = "SELECT workername,workermobileno,time from workers
w JOIN assistpincode a ON w.adminid=a.adminid where workerid='"+workerid+"';";
    ResultSet r2 = st.executeQuery(s4);
    String workername="";
    String workermobileno="";
    String time="";
    while(r2.next()){
        workername=r2.getString("workername");
        workermobileno=r2.getString("workermobileno");
        time=r2.getString("time");
        break;
    }

    conn.close();
    messageboxass m=new messageboxass();
    m.message(adminname,workername,workermobileno,time);
}
catch(Exception e){

```

```
        System.out.println("    "+e);
    }
```

```
}
```

```
    b2.doClick();
}
});
```

```
schedule.add(tfassloc);
schedule.add(tfasspin);
schedule.add(panelvehiclemodelass);
schedule.add(vehiclebrandass);
schedule.add(tfassser);
schedule.add(bres);
schedule.add(labelas);
schedule.add(panelvehiclemodel);
schedule.add(vehiclebrand);
schedule.add(paneltime);
schedule.add(paneldate);
schedule.add(panelser);
schedule.add(tfappwor);
schedule.add(tfappveh);
schedule.add(labelapp);
schedule.add(bapp);
```

```
    schedule.setVisible(true);
    dashboard.setVisible(false);
    cancel.setVisible(false);
    order.setVisible(false);
    vieworder.setVisible(false);
```

```
    }
});
```

```
b4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
```

```
b.setIcon(icon4);
b1.setIcon(icon1);
b2.setIcon(icon2);
b3.setIcon(icon3);
b4.setIcon(icon9);
```

```
vieworder.removeAll();
vieworder.revalidate();
vieworder.repaint();
vieworder.setVisible(true);
```

```
ImageIcon iconcanlb = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img47.png");
JLabel labelcan = new JLabel(iconcanlb);
labelcan.setBounds(-200,10,650,500);
```

```
Icon iconordcan = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img34.png");
JButton bordcan= new JButton(iconordcan);
bordcan.setBorderPainted(false);
bordcan.setFocusPainted(false);
bordcan.setContentAreaFilled(false);
bordcan.setBounds(110,430,250,100);
```

```
ImageIcon view = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img46.png");
JLabel labelview = new JLabel(view);
labelview.setBounds(400,50,700,500);
ArrayList<String> deliveryid=new ArrayList<String>();
deliveryid.add("-");
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
```

```
Statement st=conn.createStatement();
```

```
JTextArea ordtablet1;  
JTextArea ordtablet2;  
JTextArea ordtablet3;  
JTextArea ordtablet4;  
JTextArea ordtablet5;
```

```
String s = "SELECT orderid,no,productname,price,quantity from ecommerce  
where userid='"+userid+"' and status='not delivered'";
```

```
ResultSet r = st.executeQuery(s);  
String t1="";  
String t2="";  
String t3="";  
String t4="";  
String t5="";
```

```
ordtablet1=new JTextArea("");  
ordtablet1.setBounds(465,190,165,200);  
ordtablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
ordtablet1.setBackground(new Color(255, 222, 89));  
ordtablet1.setEditable(false);
```

```
ordtablet2=new JTextArea("");  
ordtablet2.setBounds(655,190,50,200);  
ordtablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
ordtablet2.setBackground(new Color(255, 222, 89));  
ordtablet2.setEditable(false);
```

```
ordtablet3=new JTextArea("");  
ordtablet3.setBounds(725,190,100,200);  
ordtablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
ordtablet3.setBackground(new Color(255, 222, 89));  
ordtablet3.setEditable(false);
```

```
ordtablet4=new JTextArea("");
```

```
ordtablet4.setBounds(840,190,70,200);
ordtablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
ordtablet4.setBackground(new Color(255, 222, 89));
ordtablet4.setEditable(false);
```

```
ordtablet5=new JTextArea("");
ordtablet5.setBounds(950,190,70,200);
ordtablet5.setBorder(javax.swing.BorderFactory.createEmptyBorder());
ordtablet5.setBackground(new Color(255, 222, 89));
ordtablet5.setEditable(false);
```

```
System.out.println("work1");
```

```
while(r.next()){
    deliveryid.add(r.getString("orderid"));
    t1=t1+"\n"+r.getString("orderid");
    t2=t2+"\n"+r.getString("no");
    t3=t3+"\n"+r.getString("productname");
    t4=t4+"\n"+r.getString("quantity");
    t5=t5+"\n"+r.getString("price");
}
```

```
System.out.println(t1);
System.out.println(t2);
System.out.println(t3);
System.out.println(t4);
System.out.println(t5);
```

```
ordtablet1.setText(t1);
ordtablet2.setText(t2);
ordtablet3.setText(t3);
ordtablet4.setText(t4);
ordtablet5.setText(t5);
```

```
vieworder.add(ordtablet1);
vieworder.add(ordtablet2);
vieworder.add(ordtablet3);
vieworder.add(ordtablet4);
```



```

vieworder.add(ordtablet5);

vieworder.updateUI();
conn.close();
}
catch(Exception e)
{
    System.out.println(" "+e);
}
String[] deliveryidarr=new String[deliveryid.size()];
deliveryid.toArray(deliveryidarr);

JComboBox deliveryidbox=new JComboBox(deliveryidarr);
deliveryidbox.setBounds(240,120,160,30);
deliveryidbox.setBorder(javax.swing.BorderFactory.createEmptyBorder());
deliveryidbox.setBackground(Color.white);

JPanel deliverproductboxpan=new JPanel();
deliverproductboxpan.setBounds(240,230,120,30);
deliverproductboxpan.setLayout(null);
deliverproductboxpan.setBackground(Color.white);

deliveryidbox.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent ie) {
        //To change body of generated methods, choose Tools | Templates.
        Component[] components=deliverproductboxpan.getComponents();
        for(Component component :components){

            deliverproductboxpan.remove(component);
        }
        deliverproductboxpan.revalidate();
        deliverproductboxpan.repaint();
        String deliveryid=deliveryidbox.getSelectedItem().toString();
        deliverproductboxpan.setVisible(false);
        ArrayList<String> products=new ArrayList<String>();
        products.add("-");
        try{
            Class.forName("com.mysql.jdbc.Driver");

```

```

        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String sql = "SELECT productname from ecommerce where
orderid='"+deliveryid+"'";
        ResultSet rs  = st.executeQuery(sql);
        while(rs.next()){
            products.add(rs.getString("productname"));
        }
        System.out.println("sucess district");
        System.out.println(products);
        conn.close();
    }
    catch(Exception e){
        System.out.println(" "+e);
    }
    String[] productsarr=new String[products.size()];
    products.toArray(productsarr);
    JComboBox productnamebox=new JComboBox(productsarr);
    productnamebox.setBounds(0,0,120,30);
    productnamebox.setBackground(Color.white);

    productnamebox.addItemListener(new ItemListener() {
        @Override
        public void itemStateChanged(ItemEvent ie) {
            //To change body of generated methods, choose Tools |

```

Templates.

```

productnameinputcan=productnamebox.getSelectedItem().toString();
        System.out.println(productnameinputcan);
    }
});

deliverproductboxpan.setBackground(Color.white);
deliverproductboxpan.add(productnamebox);
deliverproductboxpan.updateUI();
deliverproductboxpan.setVisible(true);
vieworder.updateUI();

```

```
}  
});
```

```
bordcan.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
        String orderid=deliveryidbox.getSelectedItemId().toString();  
        String reason=productnameinputcan;  
  
        if(orderid.isEmpty() || orderid.equals('-')){  
            Error e=new Error();  
            e.errordis("Select Deliveryid id","Check again");  
        }  
        else if(reason.isEmpty()){  
            Error e=new Error();  
            e.errordis("Select product","Pls");  
        }  
        else{  
  
            try{  
                Class.forName("com.mysql.jdbc.Driver");  
  
                Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram  
03");  
  
                Statement st=conn.createStatement();  
                String s="UPDATE ecommerce set status='cancelled' where  
orderid='"+orderid+"' and productname='"+productnameinputcan+"'";  
                st.executeUpdate(s);  
                System.out.println("Success!");  
                messageplacecancel m=new messageplacecancel();  
                m.message();  
                conn.close();  
            }  
            catch(Exception e){  
                System.out.println(" "+e);  
            }  
        }  
    }  
}
```

```
        b4.doClick();
    }
});
```

```
vieworder.add(deliverproductboxpan);
vieworder.add(deliveryidbox);
vieworder.add(bordcan);
vieworder.add(labelcan);

vieworder.add(labelview);
vieworder.updateUI();
vieworder.setVisible(true);
```

```
schedule.setVisible(false);
dashboard.setVisible(false);
cancel.setVisible(false);
order.setVisible(false);
```

```
    }
});
```

```
b3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        cancel.removeAll();
        cancel.revalidate();
        cancel.repaint();
        cancel.setVisible(true);

        b.setIcon(icon4);
        b1.setIcon(icon1);
        b2.setIcon(icon2);
        b3.setIcon(icon7);
        b4.setIcon(icon8);
```

```
Icon iconappcan = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img34.png");  
JButton bappcan= new JButton(iconappcan);  
bappcan.setBorderPainted(false);  
bappcan.setFocusPainted(false);  
bappcan.setContentAreaFilled(false);  
bappcan.setBounds(600,400,250,100);
```

```
Icon iconasscan = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img34.png");  
JButton basscan= new JButton(iconasscan);  
basscan.setBorderPainted(false);  
basscan.setFocusPainted(false);  
basscan.setContentAreaFilled(false);  
basscan.setBounds(50,400,250,100);
```

```
ImageIcon iconapplbcan = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img33.png");  
JLabel labelappcan = new JLabel(iconapplbcan);  
labelappcan.setBounds(450,10,650,500);
```

```
ImageIcon iconasslbcan = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img32.png");  
JLabel labelasscan = new JLabel(iconasslbcan);  
labelasscan.setBounds(-90,10,650,500);
```

```
ArrayList<String> applist=new ArrayList<String>();  
applist.add("-");  
try  
{  
Class.forName("com.mysql.jdbc.Driver");  
Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram  
03");  
Statement st=conn.createStatement();  
String sql = "SELECT appointmentid from appointments where  
userid='"+userid+"' and status!='Declined' and status!='OutForDelivery'";
```

```

ResultSet rs = st.executeQuery(sql);
while(rs.next()){
    applist.add(rs.getString("appointmentid"));
}
System.out.println(applist);

conn.close();
}
catch(Exception e)
{
    System.out.println(" "+e);
}

String[] apparr=new String[applist.size()];
applist.toArray(apparr);

JComboBox appidbox=new JComboBox(apparr);
appidbox.setBounds(730,200,160,30);
appidbox.setBorder(javax.swing.BorderFactory.createEmptyBorder());
appidbox.setBackground(Color.white);

JTextArea tfappreason=new JTextArea();

tfappreason.setBounds(660,280,230,50);
tfappreason.setBorder(javax.swing.BorderFactory.createEmptyBorder());
tfappreason.setBackground(Color.white);
tfappreason.setBackground(new Color(217,217,217));
tfappreason.setEditable(true);

cancel.add(tfappreason);
cancel.add(appidbox);
cancel.add(bappcan);
cancel.add(labelappcan);

bappcan.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        String appointid=appidbox.getSelectedItem().toString();
        String reason=tfappreason.getText().toString();
    }
});

```

```

        if(appointid.isEmpty() || appointid.equals('-')){
            Error e=new Error();
            e.errordis("Select appointment id","Check again");
        }
        else if(reason.isEmpty()){
            Error e=new Error();
            e.errordis("Please enter reason","Pls");
        }
        else{

            try{
                Class.forName("com.mysql.jdbc.Driver");

                Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

                Statement st=conn.createStatement();
                String s="UPDATE appointments set
status='cancelled',reason='"+reason+"' where appointmentid='"+appointid+"'";
                st.executeUpdate(s);
                System.out.println("Success!");
                messageboxappcancel m=new messageboxappcancel();
                m.message();
                conn.close();
            }
            catch(Exception e){
                System.out.println(" "+e);
            }

        }
    }
});

ArrayList<String> asslist=new ArrayList<String>();
asslist.add("-");
try
{
    Class.forName("com.mysql.jdbc.Driver");

```

```

        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st=conn.createStatement();
        String sql = "SELECT assistid from assist a  where userid='"+userid+"' and
status!='Declined' and status!='cancelled'";
        ResultSet rs  = st.executeQuery(sql);
        while(rs.next()){
            asslist.add(rs.getString("assistid"));
        }
        System.out.println(asslist);

        conn.close();
    }
    catch(Exception e)
    {
        System.out.println("    "+e);
    }

    String[] assarr=new String[asslist.size()];
    asslist.toArray(assarr);

    JComboBox assidbox=new JComboBox(assarr);
    assidbox.setBounds(240,200,160,30);
    assidbox.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    assidbox.setBackground(Color.white);

    JTextArea tfassreason=new JTextArea();

    tfassreason.setBounds(140,280,230,50);
    tfassreason.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    tfassreason.setBackground(Color.white);
    tfassreason.setBackground(new Color(217,217,217));
    tfassreason.setEditable(true);

    cancel.add(tfassreason);
    cancel.add(assidbox);
    cancel.add(basscan);
    cancel.add(labelasscan);

```



```

basscan.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        String assistid=assidbox.getSelectedItemId().toString();
        String reason=tfassreason.getText().toString();

        if(assistid.isEmpty() || assistid.equals('-')){
            Error e=new Error();
            e.errordis("Select assist id","Check again");
        }
        else if(reason.isEmpty()){
            Error e=new Error();
            e.errordis("Please enter reason","Pls");
        }
        else{

            try{
                Class.forName("com.mysql.jdbc.Driver");

                Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

                Statement st=conn.createStatement();
                String s="UPDATE assist set
status='cancelled',reason='"+reason+"' where assistid='"+assistid+"'";
                st.executeUpdate(s);
                System.out.println("Success!");
                String sql = "SELECT workerid from assist a where
assistid='"+assistid+"'";
                ResultSet rs  = st.executeQuery(sql);
                int workerid=0;
                while(rs.next()){
                    workerid=rs.getInt("workerid");
                }
                String s1="update workers set workerstatus='free' where
workerid='"+workerid+"'";
                st.executeUpdate(s1);
                messageboxasscancel m=new messageboxasscancel();
                m.message();
                conn.close();
            }
        }
    }
});

```

```

        }
        catch(Exception e){
            System.out.println(" "+e);
        }
    }
}
});

```

```

        cancel.updateUI();
        schedule.setVisible(false);
        dashboard.setVisible(false);
        cancel.setVisible(true);
        order.setVisible(false);
    }
});

```

```

frame.setLayout(null);
frame.setVisible(true);

```

```

}
public static void main(String[] args){
    dashboard d=new dashboard();
    d.dash(1);
}

```

```

public class messageboxapp {

```

```

    public void message(){

```

```

        JFrame frame1 = new JFrame();
        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame1.setUndecorated(true);

```

```

        double x = 0;
        double y = 0;
        frame1.setShape(new RoundRectangle2D.Double(x, y,400,200, 50, 50));

```

```

frame1.setSize(400, 200);
frame1.setBounds(600,250,400,200);
frame1.getContentPane().setBackground(new Color(160,237,240));

Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img25.png");
JButton b = new JButton(Icons);
b.setBorderPainted(false);
b.setFocusPainted(false);
b.setContentAreaFilled(false);
b.setBounds(190,100,200,100);
frame1.add(b);

ImageIcon icon = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img24.png");
JLabel labels = new JLabel(icon);
labels.setBounds(3,15,350,150);
frame1.add(labels);

b.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        frame1.setVisible(false);

    }
});

frame1.setLayout(null);
frame1.setVisible(true);
}

}

public class messageboxplace {

public void message(){

```

```

JFrame frame1 = new JFrame();
frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame1.setUndecorated(true);

double x = 0;
double y = 0;
frame1.setShape(new RoundRectangle2D.Double(x, y, 400, 200, 50, 50));
frame1.setSize(400, 200);
frame1.setBounds(600, 250, 400, 200);
frame1.getContentPane().setBackground(new Color(160, 237, 240));

Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img25.png");
JButton b = new JButton(Icons);
b.setBorderPainted(false);
b.setFocusPainted(false);
b.setContentAreaFilled(false);
b.setBounds(190, 100, 200, 100);
frame1.add(b);

ImageIcon icon = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img43.png");
JLabel labels = new JLabel(icon);
labels.setBounds(3, 15, 350, 150);
frame1.add(labels);

b.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        frame1.setVisible(false);

    }
});

frame1.setLayout(null);
frame1.setVisible(true);
}

```

```

}

public class messageboxappcancel {

public void message(){

    JFrame frame1 = new JFrame();
    frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame1.setUndecorated(true);

    double x = 0;
    double y = 0;
    frame1.setShape(new RoundRectangle2D.Double(x, y,400,200, 50, 50));
    frame1.setSize(400, 200);
    frame1.setBounds(600,250,400,200);
    frame1.getContentPane().setBackground(new Color(160,237,240));

    Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img25.png");
    JButton b = new JButton(icons);
    b.setBorderPainted(false);
    b.setFocusPainted(false);
    b.setContentAreaFilled(false);
    b.setBounds(190,100,200,100);
    frame1.add(b);

    ImageIcon icon = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img35.png");
    JLabel labels = new JLabel(icon);
    labels.setBounds(3,15,350,150);
    frame1.add(labels);

    b.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent ae) {

            frame1.setVisible(false);

        }
    }
}

```

```

});

frame1.setLayout(null);
frame1.setVisible(true);
}

}

public class messageboxcancel {

public void message(){

    JFrame frame1 = new JFrame();
    frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame1.setUndecorated(true);

    double x = 0;
    double y = 0;
    frame1.setShape(new RoundRectangle2D.Double(x, y,400,200, 50, 50));
    frame1.setSize(400, 200);
    frame1.setBounds(600,250,400,200);
    frame1.getContentPane().setBackground(new Color(160,237,240));

    Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img25.png");
    JButton b = new JButton(icons);
    b.setBorderPainted(false);
    b.setFocusPainted(false);
    b.setContentAreaFilled(false);
    b.setBounds(190,100,200,100);
    frame1.add(b);

    ImageIcon icon = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img36.png");
    JLabel labels = new JLabel(icon);
    labels.setBounds(3,15,350,150);
    frame1.add(labels);

    b.addActionListener(new ActionListener() {

```

```

@Override
public void actionPerformed(ActionEvent ae) {

    frame1.setVisible(false);

}

});

frame1.setLayout(null);
frame1.setVisible(true);
}

}

public class messageboxass {

public void message(String workshop,String name, String no,String time){

    JFrame frame1 = new JFrame();
    frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame1.setUndecorated(true);

    double x = 0;
    double y = 0;
    frame1.setShape(new RoundRectangle2D.Double(x, y,400,200, 50, 50));
    frame1.setSize(400, 200);
    frame1.setBounds(600,250,400,200);
    frame1.getContentPane().setBackground(new Color(160,237,240));

    JLabel ta=new JLabel(time+" mins");
    ta.setBounds(300,10,200,70);
    ta.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    ta.setForeground(Color.black);
    frame1.add(ta);

    JLabel t=new JLabel(workshop);
    t.setBounds(190,50,200,70);
    t.setBorder(javax.swing.BorderFactory.createEmptyBorder());

```

```
t.setForeground(Color.black);  
frame1.add(t);
```

```
JLabel t1=new JLabel(name);  
t1.setBounds(190,73,200,70);  
t1.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
t1.setForeground(Color.black);  
frame1.add(t1);
```

```
JLabel t2=new JLabel(no);  
t2.setBounds(190,99,200,70);  
t2.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
t2.setForeground(Color.black);  
frame1.add(t2);
```

```
Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New  
folder\\img25.png");  
JButton b = new JButton(icons);  
b.setBorderPainted(false);  
b.setFocusPainted(false);  
b.setContentAreaFilled(false);  
b.setBounds(240,115,200,100);  
frame1.add(b);
```

```
ImageIcon icon = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img28.png");  
JLabel labels = new JLabel(icon);  
labels.setBounds(3,15,350,150);  
frame1.add(labels);
```

```
b.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {  
  
        frame1.setVisible(false);
```



```

    }
    });

    frame1.setLayout(null);
    frame1.setVisible(true);
}

}

public class messageplacecancel {

    public void message(){

        JFrame frame1 = new JFrame();
        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame1.setUndecorated(true);

        double x = 0;
        double y = 0;
        frame1.setShape(new RoundRectangle2D.Double(x, y, 400, 200, 50, 50));
        frame1.setSize(400, 200);
        frame1.setBounds(600, 250, 400, 200);
        frame1.getContentPane().setBackground(new Color(160, 237, 240));

        Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img25.png");
        JButton b = new JButton(icons);
        b.setBorderPainted(false);
        b.setFocusPainted(false);
        b.setContentAreaFilled(false);
        b.setBounds(200, 110, 200, 100);
        frame1.add(b);

        ImageIcon icon = new
        ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img48.png");
        JLabel labels = new JLabel(icon);

```

```

        labels.setBounds(3,15,350,150);
        frame1.add(labels);

        b.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {

                frame1.setVisible(false);

            }
        });

        frame1.setLayout(null);
        frame1.setVisible(true);
    }

}
}

```

table.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.RoundRectangle2D;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.Icon;

```

```

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;

/**
 *
 * @author PROJECTS
 */
public class table {

    public void tables(int userid){

        JFrame frame = new JFrame();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setUndecorated(true);

        double x = 0;
        double y = 0;
        frame.setShape(new RoundRectangle2D.Double(x, y,1200,600, 50, 50));
        frame.setSize(1200, 600);
        frame.setBounds(30,30,1200,600);
        frame.getContentPane().setBackground(Color.white);

        Icon iconclose = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img56.png");
        JButton close= new JButton(iconclose);
        close.setBorderPainted(false);
        close.setFocusPainted(false);
        close.setContentAreaFilled(false);
        close.setBounds(930,500,250,100);

        frame.add(close);

        close.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent ae) {

```

```

        frame.setVisible(false);
        dashboard d=new dashboard();
        d.dash(userid);

    }
});

    Icon icon = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img50.png");
    JButton b = new JButton(icon);
    b.setBorderPainted(false);
    b.setFocusPainted(false);
    b.setContentAreaFilled(false);
    b.setBounds(-50,30,250,100);
    frame.add(b);

    Icon icon1 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img53.png");
    JButton b1 = new JButton(icon1);
    b1.setBorderPainted(false);
    b1.setFocusPainted(false);
    b1.setContentAreaFilled(false);
    b1.setBounds(-50,100,250,100);
    frame.add(b1);

    Icon icon2 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img51.png");
    Icon icon3 = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img52.png");
    ImageIcon icons = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img19.png");
    JLabel label = new JLabel(icons);
    label.setBounds(0,0,150,800);
    frame.add(label);

    JPanel assist=new JPanel();
    assist.setBackground(Color.white);

```

```
ImageIcon iconass = new  
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img54.png");  
JLabel labelass = new JLabel(iconass);  
labelass.setBounds(0,-15,900,600);
```

```
assist.add(labelass);  
assist.setLayout(null);  
assist.setBounds(150,0,1150,600);
```

```
try  
{  
Class.forName("com.mysql.jdbc.Driver");  
Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram  
03");  
Statement st=conn.createStatement();
```

```
JTextArea asstablet1;  
JTextArea asstablet2;  
JTextArea asstablet3;  
JTextArea asstablet4;
```

```
String s = "select assistid,name,timestamp,status from assist a join workers w ON  
a.workerid=w.workerid join admin ad ON ad.adminid=w.adminid where  
userid='"+userid+"'";
```

```
ResultSet r = st.executeQuery(s);  
String t1="";  
String t2="";  
String t3="";  
String t4="";
```

```
asstablet1=new JTextArea("");  
asstablet1.setBounds(100,120,20,400);  
asstablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
asstablet1.setBackground(new Color(255, 229, 127));  
asstablet1.setEditable(false);
```

```
asstablet2=new JTextArea("");
asstablet2.setBounds(190,120,150,400);
asstablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet2.setBackground(new Color(255, 229, 127));
asstablet2.setEditable(false);
```

```
asstablet3=new JTextArea("");
asstablet3.setBounds(420,120,230,400);
asstablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet3.setBackground(new Color(255, 229, 127));
asstablet3.setEditable(false);
```

```
asstablet4=new JTextArea("");
asstablet4.setBounds(710,120,70,400);
asstablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
asstablet4.setBackground(new Color(255, 229, 127));
asstablet4.setEditable(false);
```

```
while(r.next()){
    t1=t1+"\n"+r.getString("assistid");
    t2=t2+"\n"+r.getString("timestamp");
    t3=t3+"\n"+r.getString("name");
    t4=t4+"\n"+r.getString("status");
}
asstablet1.setText(t1);
asstablet2.setText(t2);
asstablet3.setText(t3);
asstablet4.setText(t4);
```

```
assist.add(asstablet1);
assist.add(asstablet2);
assist.add(asstablet3);
assist.add(asstablet4);
```

```
conn.close();  
}  
catch(Exception e)  
{  
    System.out.println("    "+e);  
}
```

```
assist.add(labelass);
```

```
assist.setVisible(true);  
frame.add(assist);
```

```
JPanel appointment=new JPa nel();  
appointment.setBackground(Color.white);
```

```
appointment.setLayout(null);  
appointment.setBounds(150,0,1150,600);
```

```
appointment.setVisible(true);  
frame.add(appointment);
```

```
b.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent ae) {
```

```
        assist.removeAll();  
        assist.revalidate();  
        assist.repaint();  
        assist.setVisible(true);
```

```
        b.setIcon(icon);  
        b1.setIcon(icon1);
```

```

        ImageIcon iconass = new
        ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img54.png");
        JLabel labelass = new JLabel(iconass);
        labelass.setBounds(0,-15,900,600);

        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
            03");
            Statement st=conn.createStatement();

            JTextArea asstablet1;
            JTextArea asstablet2;
            JTextArea asstablet3;
            JTextArea asstablet4;

            String s = "select assistid,name,timestamp,status from assist a join workers w ON
            a.workerid=w.workerid join admin ad ON ad.adminid=w.adminid where
            userid='"+userid+"'";
            ResultSet r  = st.executeQuery(s);
            String t1="";
            String t2="";
            String t3="";
            String t4="";

            asstablet1=new JTextArea("");
            asstablet1.setBounds(100,120,20,400);
            asstablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
            asstablet1.setBackground(new Color(255, 229, 127));
            asstablet1.setEditable(false);

            asstablet2=new JTextArea("");
            asstablet2.setBounds(190,120,150,400);
            asstablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
            asstablet2.setBackground(new Color(255, 229, 127));

```



```
asstablet2.setEditable(false);
```

```
asstablet3=new JTextArea("");  
asstablet3.setBounds(420,120,230,400);  
asstablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
asstablet3.setBackground(new Color(255, 229, 127));  
asstablet3.setEditable(false);
```

```
asstablet4=new JTextArea("");  
asstablet4.setBounds(710,120,70,400);  
asstablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());  
asstablet4.setBackground(new Color(255, 229, 127));  
asstablet4.setEditable(false);
```

```
while(r.next()){  
    t1=t1+"\n"+r.getString("assistid");  
    t2=t2+"\n"+r.getString("timestamp");  
    t3=t3+"\n"+r.getString("name");  
    t4=t4+"\n"+r.getString("status");  
}  
asstablet1.setText(t1);  
asstablet2.setText(t2);  
asstablet3.setText(t3);  
asstablet4.setText(t4);
```

```
assist.add(asstablet1);  
assist.add(asstablet2);  
assist.add(asstablet3);  
assist.add(asstablet4);
```

```
conn.close();  
}  
catch(Exception e)  
{  
    System.out.println(" "+e);  
}
```

```

        assist.add(labelass);

        assist.updateUI();
        assist.setVisible(true);
        appointment.setVisible(false);

    }
});

b1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {

        b.setIcon(icon2);
        b1.setIcon(icon3);
        appointment.removeAll();
        appointment.revalidate();
        appointment.repaint();
        appointment.setVisible(true);

        ImageIcon iconapp = new
        ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img55.png");
        JLabel labelapp = new JLabel(iconapp);
        labelapp.setBounds(0,-15,900,600);

        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
            03");
            Statement st=conn.createStatement();

            JTextArea apptablet1;
            JTextArea apptablet2;
            JTextArea apptablet3;
            JTextArea apptablet4;

            String s = "select appointmentid,name,id,date,status from appointments a join
            admin ad ON ad.adminid=a.adminid where userid='"+userid+"'";

```

```
ResultSet r = st.executeQuery(s);
String t1="";
String t2="";
String t3="";
String t4="";
```

```
apptablet1=new JTextArea("");
apptablet1.setBounds(100,120,20,400);
apptablet1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet1.setBackground(new Color(255, 229, 127));
apptablet1.setEditable(false);
```

```
apptablet2=new JTextArea("");
apptablet2.setBounds(190,120,150,400);
apptablet2.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet2.setBackground(new Color(255, 229, 127));
apptablet2.setEditable(false);
```

```
apptablet3=new JTextArea("");
apptablet3.setBounds(420,120,230,400);
apptablet3.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet3.setBackground(new Color(255, 229, 127));
apptablet3.setEditable(false);
```

```
apptablet4=new JTextArea("");
apptablet4.setBounds(710,120,70,400);
apptablet4.setBorder(javax.swing.BorderFactory.createEmptyBorder());
apptablet4.setBackground(new Color(255, 229, 127));
apptablet4.setEditable(false);
```

```
while(r.next()){
    t1=t1+"\n"+r.getString("appointmentid");
    t2=t2+"\n"+r.getString("date")+" "+r.getString("id").substring(0,8);
    t3=t3+"\n"+r.getString("name");
    t4=t4+"\n"+r.getString("status");
}
```

```

        apptablet1.setText(t1);
        apptablet2.setText(t2);
        apptablet3.setText(t3);
        apptablet4.setText(t4);

        appointment.add(apptablet1);
        appointment.add(apptablet2);
        appointment.add(apptablet3);
        appointment.add(apptablet4);

        conn.close();
    }
    catch(Exception e)
    {
        System.out.println(" "+e);
    }

        appointment.add(labelass);

        appointment.updateUI();
        appointment.setVisible(true);
        assist.setVisible(false);

    }

});

frame.setLayout(null);
frame.setVisible(true);

}

public static void main(String[] args){

    table t=new table();
    t.tables(1);
}
}

```

Error.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package User;
/**
 *
 * @author PROJECTS
 */
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.RoundRectangle2D;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
/**
 *
 * @author PROJECTS
 */
public class Error {
    public void errorDis(String s,String a){
        JFrame frame1 = new JFrame();
        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame1.setUndecorated(true);

        double x = 0;
        double y = 0;
        frame1.setShape(new RoundRectangle2D.Double(x, y,400,200, 50, 50));
        frame1.setSize(400, 200);
        frame1.setBounds(600,250,400,200);
        frame1.getContentPane().setBackground(Color.white);

        JLabel t=new JLabel(s);
        t.setBounds(140,50,200,70);
        t.setBorder(javax.swing.BorderFactory.createEmptyBorder());
    }
}
```

```

t.setForeground(Color.red);
frame1.add(t);

JLabel t1=new JLabel(a);
t1.setBounds(140,70,200,70);
t1.setBorder(javax.swing.BorderFactory.createEmptyBorder());
t1.setForeground(Color.red);
frame1.add(t1);

Icon icons = new ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New
folder\\img27.png");
JButton b = new JButton(icons);
b.setBorderPainted(false);
b.setFocusPainted(false);
b.setContentAreaFilled(false);
b.setBounds(190,100,200,100);
frame1.add(b);

ImageIcon icon = new
ImageIcon("C:\\Users\\PROJECTS\\Desktop\\projects\\New folder\\img26.png");
JLabel labels = new JLabel(icon);
labels.setBounds(3,15,350,150);
frame1.add(labels);

b.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent ae) {
        frame1.setVisible(false);
    }
});

frame1.setLayout(null);
frame1.setVisible(true);

}
public static void main(String[] args){
    Error e=new Error();
    e.errordis("Choose another time","It's already booked");
}
}

```

ADMIN MODULE

Main Page.java

```
package Admin;

import java.awt.Button;
import java.awt.Color;
import java.awt.Component;
import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.Label;
import java.awt.TextField;
import java.awt.FlowLayout;
import java.awt.Frame;
import java.awt.Image;
import javax.swing.ImageIcon;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import java.awt.event.TextListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.*.*;

public class MainPage extends JFrame
{
    JButton log,reg;
    JPanel contentPane;
    JTextPane p;
    JLabel label,a;
    MainPage()
    {
        setLayout(null);
        Container c=getContentPane();
        contentPane = new JPanel();
        log= new JButton("Login");
        log.setFont(new Font("Courier New", Font.BOLD, 15));

        reg= new JButton("Register");
```

```

reg.setFont(new Font("Courier New", Font.BOLD, 15));

log.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
new Login();
dispose();
}
});
reg.addActionListener((ActionEvent e) -> {
new MyFrame();
dispose();

});
Frame f=new JFrame();
Image icon = Toolkit.getDefaultToolkit().getImage("C:\\\\AUTO CARE\\\\admin-
images.jpg");
setTitle("Admin Login Page");
setIconImage(icon);
ImageIcon imagelcon = new ImageIcon("C:\\\\AUTO CARE\\\\by-my-car-illustration-
concept.jpg");
label = new JLabel(imagelcon);
label.setBounds(5,40,626,417);
add(label);
a=new JLabel();
a.setText("AUTO-CARE");
a.setBounds(00,00,1376,768);
a.setFont(new Font("TIMES NEW ROMAN", Font.BOLD, 25));

add(a);
log.setBounds(650,200,200,30);
log.setForeground(Color.BLACK);
add(log);
reg.setBounds(650,300,200,30);
reg.setForeground(Color.BLACK);
add(reg);

setSize(900,500);
setVisible(true);
c.setBackground(Color.white);
addWindowListener(new WindowAdapter(){

```



```

@Override
public void windowClosing(WindowEvent e) {
    dispose();
}
});
}
public static void main(String[] args)
{
    JFrame.setDefaultLookAndFeelDecorated(true);
    new MainPage();

}
}

```

Myframe.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package Admin;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import static javafx.scene.paint.Color.color;

class MyFrame extends JFrame {

    private Container c;
    private JLabel title;
    private JLabel name;
    private JTextField tname;
    private JLabel mno;
    private JTextField tmno;

```

```
private JLabel city;
private JTextField tcity;
private JLabel state;
private JTextField tstate;
private JLabel Admin_name;
private JTextField tAdmin_name;
private JLabel pass;
private JTextField tpass;
private JLabel pass2;
private JTextField tpass2;
private JLabel pincode;
private JTextField tpincode;
private JLabel add1,add2;
private JTextField tadd1,tadd2;
private JCheckBox term;
private JButton sub;
private JButton reset;
private JTextArea tout;
private JLabel res;
private JTextArea resadd;

public MyFrame()
{
    Color pwhite=new Color( 244,248,247);
    setTitle("Admin Registration Form");
    setBounds(0, 0, 1366, 768);
    setExtendedState(JFrame.MAXIMIZED_BOTH);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    //setResizable(false);

    Color white=new Color(255,255,255);
    Color blue=new Color(0,98,205);
    Color hblue=new Color(34,72,102);

    c = getContentPane();
    c.setLayout(null);
    c.setBackground(white);

    title = new JLabel("Admin Registration Form");
    title.setFont(new Font("bold", Font.PLAIN, 30));
```

```
title.setSize(370, 40);
```

```
title.setLocation(140, 30);  
c.add(title);
```

```
Admin_name = new JLabel("Workshop Name");  
Admin_name.setFont(new Font("Arial", Font.PLAIN, 20));  
Admin_name.setSize(200, 20);  
Admin_name.setLocation(60, 100);  
Admin_name.setForeground(hblue);  
c.add(Admin_name);
```

```
tAdmin_name = new JTextField();  
tAdmin_name.setFont(new Font("Arial", Font.PLAIN, 15));  
tAdmin_name.setSize(190, 20);  
tAdmin_name.setLocation(250, 100);  
tAdmin_name.setBackground(pwhite);  
c.add(tAdmin_name);
```

```
name = new JLabel("Workshop Username");  
name.setFont(new Font("Arial", Font.PLAIN, 20));  
name.setSize(250, 20);  
name.setForeground(hblue);  
name.setLocation(60, 150);  
c.add(name);
```

```
tname = new JTextField();  
tname.setFont(new Font("Arial", Font.PLAIN, 15));  
tname.setSize(190, 20);  
tname.setLocation(250, 150);  
tname.setBackground(pwhite);  
c.add(tname);
```

```
mno = new JLabel("Mobile Number");  
mno.setFont(new Font("Arial", Font.PLAIN, 20));  
mno.setSize(200, 20);  
mno.setForeground(hblue);  
mno.setLocation(60, 200);  
c.add(mno);
```

```
tmno = new JTextField();
```

```
tmno.setFont(new Font("Arial", Font.PLAIN, 15));
tmno.setSize(190, 20);
tmno.setLocation(250, 200);
tmno.setBackground(pwhite);
c.add(tmno);
```

```
pass = new JLabel("Password");
pass.setFont(new Font("Arial", Font.PLAIN, 20));
pass.setSize(200, 20);
pass.setForeground(hblue);
pass.setLocation(60, 250);
c.add(pass);
```

```
tpass = new JTextField();
tpass.setFont(new Font("Arial", Font.PLAIN, 15));
tpass.setSize(190, 20);
tpass.setBackground(pwhite);
tpass.setLocation(250, 250);
c.add(tpass);
```

```
pass2 = new JLabel("Re-Enter Password");
pass2.setFont(new Font("Arial", Font.PLAIN, 20));
pass2.setSize(200, 20);
pass2.setForeground(hblue);
pass2.setLocation(60, 300);
c.add(pass2);
```

```
tpass2 = new JTextField();
tpass2.setFont(new Font("Arial", Font.PLAIN, 15));
tpass2.setSize(190, 20);
tpass2.setLocation(250, 300);
tpass2.setBackground(pwhite);
c.add(tpass2);
```

```
add1 = new JLabel("Address line 1");
add1.setFont(new Font("Arial", Font.PLAIN, 20));
add1.setSize(250, 20);
add1.setForeground(hblue);
add1.setLocation(60, 350);
c.add(add1);
```

```
tadd1 = new JTextField();  
tadd1.setFont(new Font("Arial", Font.PLAIN, 15));  
tadd1.setSize(250, 20);  
tadd1.setLocation(250, 350);  
tadd1.setBackground(pwhite);  
c.add(tadd1);
```

```
add2 = new JLabel("Address line 2");  
add2.setFont(new Font("Arial", Font.PLAIN, 20));  
add2.setSize(250, 20);  
add2.setForeground(hblue);  
add2.setLocation(60, 400);  
c.add(add2);
```

```
tadd2 = new JTextField();  
tadd2.setFont(new Font("Arial", Font.PLAIN, 15));  
tadd2.setSize(250, 20);  
tadd2.setBackground(pwhite);  
tadd2.setLocation(250, 400);  
c.add(tadd2);
```

```
city = new JLabel("city");  
city.setFont(new Font("Arial", Font.PLAIN, 20));  
city.setSize(100, 20);  
city.setForeground(hblue);  
city.setLocation(60, 450);  
c.add(city);
```

```
tcity = new JTextField();  
tcity.setFont(new Font("Arial", Font.PLAIN, 15));  
tcity.setSize(190, 20);  
tcity.setBackground(pwhite);  
tcity.setLocation(250, 450);  
c.add(tcity);
```

```
state = new JLabel("state");  
state.setFont(new Font("Arial", Font.PLAIN, 20));  
state.setSize(100, 20);  
state.setForeground(hblue);  
state.setLocation(60, 500);
```

```
c.add(state);
```

```
tstate = new JTextField();  
tstate.setFont(new Font("Arial", Font.PLAIN, 15));  
tstate.setSize(190, 20);  
tstate.setBackground(pwhite);  
tstate.setLocation(250, 500);  
c.add(tstate);
```

```
pincode = new JLabel("pincode");  
pincode.setFont(new Font("Arial", Font.PLAIN, 20));  
pincode.setSize(100, 20);  
pincode.setForeground(hblue);  
pincode.setLocation(60, 550);  
c.add(pincode);
```

```
tpincode = new JTextField();  
tpincode.setFont(new Font("Arial", Font.PLAIN, 15));  
tpincode.setSize(190, 20);  
tpincode.setBackground(pwhite);  
tpincode.setLocation(250, 550);  
c.add(tpincode);
```

```
term = new JCheckBox("Accept Terms And Conditions.");  
term.setFont(new Font("bold", Font.PLAIN, 15));  
term.setSize(250, 20);  
term.setBackground(white);  
term.setLocation(250, 600);  
c.add(term);
```

```
sub = new JButton("Register");  
sub.setFont(new Font("Arial", Font.PLAIN, 15));  
sub.setForeground(white);  
sub.setSize(100, 20);  
sub.setBackground(blue);  
sub.setLocation(250, 650);  
sub.addActionListener(new ActionListener()  
{  
    public void actionPerformed(ActionEvent e)
```



```

        {
            JOptionPane.showMessageDialog(rootPane,"Entered Mobile
Number Does'nt exist");
        }
        else
        {
            if(tpincode.getText().length()!=6)
            {
                JOptionPane.showMessageDialog(rootPane,"Pincode does not
exist");
            }
            else
            {
                ps.executeUpdate();
                JOptionPane.showMessageDialog(rootPane,"You have
Registered Successfully");
                new Login();
            }
        }
    }

    }
    else
    {
        JOptionPane.showMessageDialog(rootPane,"Username already
taken");
    }
    con.close();
}
catch(Exception exp)
{
    JOptionPane.showMessageDialog(rootPane,exp);
}
}
else
{
    JOptionPane.showMessageDialog(rootPane,"Passwords do not
match");
}
}
}

```



```

        else
        {
            JOptionPane.showMessageDialog(rootPane,"Accept Our Terms and
Conditions");
        }
    }
});
c.add(sub);
setBounds(0,0,600,768);
setVisible(true);
}

}

class JavaApplication14 {

    public static void main(String[] args)
    {
        new MyFrame();
    }
}

```

Login.java

```

package Admin;
import static com.sun.java.accessibility.util.AWTEventMonitor.addTextListener;
import javax.swing.*.*;
import java.awt.*.*;
import javax.swing.border.EmptyBorder;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import java.awt.event.TextListener;
import javax.swing.ImageIcon;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```
import sun.security.util.Password;

class Login extends JFrame{
public static int adminid;
JLabel p,a;
JLabel l1,l2;
JTextField t1;
JPasswordField t2;
JButton l,b;
JPanel contentPane;
Login(){

setLayout(null);
Container c=getContentPane();
contentPane = new JPanel();
Image icon = Toolkit.getDefaultToolkit().getImage("C:\\\\AUTO CARE\\\\admin-
images.jpg");
setTitle("Admin Login Page");
setIconImage(icon);
l1=new JLabel("Admin Name");
l2=new JLabel("Password");

t1=new JTextField();

t1.setToolTipText("Enter admin name ");
t2=new JPasswordField();
addTextListener(new TextListener() {
@Override
public void textValueChanged(TextEvent e) {
String s1=t1.getText();
char[] password = t2.getPassword();
t2.setEchoChar();
}
});
l=new JButton("Login");
l.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
try {
Class.forName("com.mysql.jdbc.Driver");
Connection connection = (Connection)
```

```

DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
PreparedStatement st= (PreparedStatement) connection.prepareStatement("Select *
from Admin where adminname=? ");
String s1=t1.getText();

st.setString(1, s1);
String s2=new String(t2.getPassword());
ResultSet rs = st.executeQuery();
if(rs.next())
{
if(s2.equals(rs.getString(4)))
{
JOptionPane.showMessageDialog(l,"You have successfully logged in");
adminid=rs.getInt("adminid");
dispose();
new Profile();
}
else
{
JOptionPane.showMessageDialog(l,"username Password Does not Match");
}
}
else
{
JOptionPane.showMessageDialog(l,"Register to use our application");
}
}
catch (Exception ae)
{
System.out.println(" "+ae);
}
}
});
b=new JButton("Back");
b.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
new MainPage();
dispose();
}
}

```

```

});

l1.setBounds(100,120,100,25);
l1.setFont(new Font("Times New Roman",Font.BOLD,16));
contentPane.add(l1);

l2.setBounds(100,170,100,25);
l2.setFont(new Font("Times New Roman",Font.BOLD,16));

t1.setBounds(230,120,150,30);
t1.setFont(new Font("Candara", Font.BOLD, 14));
t1.setBackground(Color.LIGHT_GRAY);
t2.setBounds(230,170,150,30);
t2.setBackground(Color.LIGHT_GRAY);
l.setBounds(290,250,80,25);
l.setFont(new Font("Courier New", Font.BOLD, 14));
b.setBounds(150,250,80,25);
b.setFont(new Font("Courier New", Font.BOLD, 14));
Imagelcon imagelcon = new Imagelcon("C:\\AUTO CARE\\by-my-car-illustration-
concept.jpg");
p= new JLabel(imagelcon);
p.setBounds(390,5,626,417);

add(l1);
add(l2);
add(t1);
add(t2);
add(l);
add(b);
add(p);
setSize(1020,500);
c.setBackground(Color.white);
setResizable(true);
setVisible(true);
}
public static void main(String args[]){
JFrame.setDefaultLookAndFeelDecorated(true);
new Login();
}
}

```

Admin.java

```
package Admin;

import static Admin.Login.*;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableCellRenderer;
//Immediate Assistance TAb
public class Admin extends JFrame{
    //Defining default AdminId to be 1:
    //int AID = adminid;
    int AID=1;
    JTabbedPane tp;
    JPanel p1;
    JButton ref2,b_accept,b_decline,b_respond;
    JLabel l_userID,l_min,l_inmin;
    JTextField tf_userID;
    JComboBox cb_min;
    String ID;
    int count;

    Admin(){
```

```
tp = new JTabbedPane();
tp.setBounds(0,0,1366, 768);
p1 = new JPanel();
p1.setLayout(null);

String icon1 = "/resources/immediate 1.png";
ImageIcon icon = createImageIcon(icon1);

tp.addTab("Immediate Assistance",icon,p1,"Immediate Assistane");
add(tp);
tp.setBackgroundAt(0,Color.WHITE);
p1.setBackground(Color.WHITE);

new ShowTable();

String icon2 = "/resources/refresh 2.jpg";
ImageIcon b_icon = createImageIcon(icon2);

ref2 = new JButton(b_icon);
ref2.setText("REFRESH");
ref2.setBounds(1100, 100, 120, 40);
ref2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        new ShowTable();
    }
});
p1.add(ref2);
ref2.setBackground(Color.WHITE);

l_userID = new JLabel("AssistID");
l_userID.setBounds(1050,200,80,40);
p1.add(l_userID);

tf_userID = new JTextField();
tf_userID.setBounds(1130,200,150,40);
p1.add(tf_userID);

l_min = new JLabel("Estimated Time to Reach The Location :");
```

```

l_min.setBounds(1050,400,500,40);
p1.add(l_min);
l_min.setVisible(false);

String minutes[] = {"10","20","30","40"};
cb_min = new JComboBox(minutes);
cb_min.setBounds(1070,450,80,40);
p1.add(cb_min);
cb_min.setVisible(false);

l_inmin = new JLabel("In Minutes.");
l_inmin.setBounds(1170,450,100,40);
p1.add(l_inmin);
l_inmin.setVisible(false);

b_accept = new JButton("Accept");
b_accept.setBounds(1060,250,90,40);
p1.add(b_accept);
b_accept.setBackground(Color.WHITE);
b_accept.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){

        ID = tf_userID.getText();
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update assist set status = 'Approved' where assistid =
""+ID+"" and status!='Declined'";
            st.executeUpdate(query);
            String query1= "select * from assist where assistid = ""+ID+"" and
status!='Declined'";
            ResultSet rs=st.executeQuery(query1);
            if(rs.next())
            {
                l_min.setVisible(true);
                cb_min.setVisible(true);

```

```

        l_inmin.setVisible(true);
        b_respond.setVisible(true);
        new ShowTable();
    }
    con.close();
    new ShowTable();
}
catch(Exception ex){
    JOptionPane.showMessageDialog(rootPane, ex);
}
}
});

b_decline = new JButton("Decline");
b_decline.setBounds(1160,250,90,40);
p1.add(b_decline);
b_decline.setBackground(Color.WHITE);
b_decline.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update assist set status = 'Declined' where assistid =
""+ID+""";
            st.executeUpdate(query);
            con.close();
            new ShowTable();
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane, ex);
        }
    }
});
//Responding Estimated time to user
b_respond = new JButton("RESPOND USER");
b_respond.setBounds(1050,500,200,40);
p1.add(b_respond);
b_respond.setBackground(Color.WHITE);

```



```

b_respond.setVisible(false);
b_respond.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try{
            ID = tf_userID.getText();
            Object minutes = cb_min.getItemAt(cb_min.getSelectedIndex());
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update assist set EstimatedTime = '"+minutes+"' where
assistid = '"+ID+"'";
            st.executeUpdate(query);
            con.close();
            new ShowTable();
            JOptionPane.showMessageDialog(rootPane,"EstimatedTime Updated to
"+minutes+" minutes");
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane, ex);
        }
    }
});

//setSize(1170,1000);
setLayout(null);
setLayout(null);
setVisible(true);
setExtendedState(JFrame.MAXIMIZED_BOTH);
}

//class finding the number of records in a DataBase Table
public class Count{
    Count(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

```

```

        Statement st = con.createStatement();
        String query = "select count(*) from assist where adminid='"+AID+"'";
        ResultSet rs = st.executeQuery(query);
        rs.next();
        count = rs.getInt(1);
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane,ex);
    }
}
}
}

```

//Class Displaying Table In Pannel

```

public class ShowTable {
    String data[][];
    String colname[];
    JTable jt1;
    ShowTable(){
        new Count();
        data = new String[count][12];
        String [] heading = {"Assist ID","UserID","Admin Id","Worker
ID","landmark","Pincode","VehicleBrand","VehicleModel","Service","status","Estima
tedTime","Reason"};

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "select * from assist where adminid='"+AID+"'";
            ResultSet rs = st.executeQuery(query);

            int row = 0;
            while(rs.next()) {
                data[row][0] = rs.getString("assistid");
                data[row][1] = rs.getString("userid");
                data[row][2] = rs.getString("adminid");
                data[row][3] = rs.getString("workerid");
                data[row][6] = rs.getString("vehiclebrand");
                data[row][7] = rs.getString("vehiclemodel");
            }
        }
    }
}

```

```

        data[row][4] = rs.getString("landmark");
        data[row][5] = rs.getString("pincode");
        data[row][8] = rs.getString("service");
        data[row][9] = rs.getString("status");
        data[row][10] = rs.getString("EstimatedTime");
        data[row][11] = rs.getString("reason");
        row++;
    }
    jt1 = new JTable(data,heading);
    jt1.setBounds(10,30,1000,600);

    jt1.setEnabled(false);

    JScrollPane jsp = new JScrollPane(jt1);
    jsp.setBounds(10,38,1014,612);

    int height = jt1.getRowHeight();
    jt1.setRowHeight(height+35);
    //Aligning the text in JTable to center
    DefaultTableCellRenderer centerRenderer = new
    DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment( JLabel.CENTER );
    for(int i=0;i<12;i++){
        jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
    }

    p1.add(jsp);
    con.close();
}
catch(Exception e) {
    JOptionPane.showMessageDialog(p1,e.toString());
}
jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
jt1.setLayout(null);

}
}

protected static ImageIcon createmageIcon(String path) {
    java.net.URL imgURL = Admin.class.getResource(path);

```

```

        if (imgURL != null) {
            return new ImageIcon(imgURL);
        } else {
            System.err.println("Couldn't find file: " + path);
            return null;
        }
    }
}

public static void main(String[] args) {
    new Admin();
}
}

```

Viewbooking.java

```

package Admin;

import static Admin.Admin.createImageIcon;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableCellRenderer;

public class ViewBooking extends Admin{
    JPanel p2;
    JButton ref3;
}

```

```
JLabel l_userid,l_status;
JTextField tf_userid;
JButton accept,decline,respondstatus;
JComboBox cb_status;
String status,ID;
ViewBooking(){
    p2 = new JPanel();

    String icont1 = "/resources/viewbook.png";
    ImageIcon icon = createImageIcon(icont1);

    tp.addTab("View Booking",icon,p2,"Manage Bookings");
    p2.setLayout(null);
    tp.setBackgroundAt(1,Color.WHITE);
    p2.setBackground(Color.WHITE);

    new ShowTable();

    String icont2 = "/resources/refresh 2.jpg";
    ImageIcon b_icon = createImageIcon(icont2);

    ref3 = new JButton(b_icon);
    ref3.setText("REFRESH");
    ref3.setBounds(1100, 100, 120, 40);
    ref3.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
            new ShowTable();
        }
    });
    p2.add(ref3);
    ref3.setBackground(Color.WHITE);

    l_userid = new JLabel("AppointmentID ");
    l_userid.setBounds(1050,200,110,40);
    p2.add(l_userid);

    tf_userid = new JTextField();
    tf_userid.setBounds(1160,200,110,40);
    p2.add(tf_userid);

    l_status = new JLabel("Status");
```

```

l_status.setBounds(1050,400,70,40);
p2.add(l_status);
l_status.setVisible(false);

String status[] = {"OutForPickup","VehiclePicked","OutForDelivery"};
cb_status = new JComboBox<String>(status);
cb_status.setBounds(1130,400,150,40);
p2.add(cb_status);
cb_status.setVisible(false);

String icon3 = "/resources/updatesstatus.png";
ImageIcon b_icon2 = new ImageIcon(icon3);

respondstatus = new JButton(b_icon2);
respondstatus.setText("UPDATE STATUS");
respondstatus.setBounds(1060,450,180,40);
p2.add(respondstatus);
respondstatus.setBackground(Color.WHITE);
respondstatus.setVisible(false);
respondstatus.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try{
            Object status = cb_status.getItemAt(cb_status.getSelectedIndex());
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update appointments set status = '"+status+"' where
appointmentid = '"+ID+"'";
            st.executeUpdate(query);
            con.close();
            new ShowTable();
            JOptionPane.showMessageDialog(rootPane,"Status Updated");
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane, ex);
        }
    }
});

```

```

accept = new JButton("Accept");
accept.setBounds(1060,280,90,40);
p2.add(accept);
accept.setBackground(Color.WHITE);
accept.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        ID = tf_userid.getText();
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update appointments set status = 'Booked' where
appointmentid = '"+ID+"'";
            st.executeUpdate(query);
            con.close();
            new ShowTable();
        }
        catch(Exception ex){
            System.out.println(ex);
        }
        l_status.setVisible(true);
        cb_status.setVisible(true);
        respondstatus.setVisible(true);
    }
});

decline = new JButton("Decline");
decline.setBounds(1160,280,90,40);
p2.add(decline);
decline.setBackground(Color.WHITE);
decline.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try{
            ID = tf_userid.getText();
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();

```

```

        String query = "update appointments set status = 'Declined' where
appointmentid = '"+ID+"'";
        st.executeUpdate(query);
        con.close();
        new ShowTable();
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane, ex);
    }
}
});
}

```

//class finding the number of records in a DataBase Table

```

public class Count{
    Count(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
            Statement st = con.createStatement();
            String query = "select count(*) from appointments where adminid =
 '"+AID+"'";
            ResultSet rs = st.executeQuery(query);
            rs.next();
            count = rs.getInt(1);
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane,ex);
        }
    }
}

```

```

public class ShowTable {
    String [][]data;
    String[]col name;
    JTable jt1;
    ShowTable(){
        new Count();
    }
}

```



```

        data = new String[count][8];
        String [] heading =
{"UserID","VehicleNo","AppointmentID","Service","Status","Time And
Date","VehicleBrand","VehicleModel"};

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "select * from appointments where adminid = '"+AID+"'";
            ResultSet rs = st.executeQuery(query);

            int row = 0;
            while(rs.next()) {
                data[row][0] = rs.getString("userid");
                data[row][1] = rs.getString("vehicleno");
                data[row][2] = rs.getString("appointmentid");
                data[row][3] = rs.getString("service");
                data[row][4] = rs.getString("status");
                data[row][5] = rs.getString("id");
                data[row][6] = rs.getString("vehiclebrand");
                data[row][7] = rs.getString("vehiclemodel");

                row++;
            }
            jt1 = new JTable(data,heading);
            jt1.setBounds(10,30,1000,600);

            jt1.setEnabled(false);

            JScrollPane jsp = new JScrollPane(jt1);
            jsp.setBounds(10,38,1014,612);

            int height = jt1.getRowHeight();
            jt1.setRowHeight(height+35);

//Aligning the text in JTable to center
            DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();

```

```

        centerRenderer.setHorizontalAlignment( JLabel.CENTER );
        for(int i=0;i<8;i++){
            jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
        }

        p2.add(jsp);
        con.close();
    }
    catch(Exception e) {
        JOptionPane.showMessageDialog(p2,e.toString());
    }
    jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
    jt1.setLayout(null);

    }
}

public static void main(String args[]){
    new ViewBooking();
}

}

```

Partsreq.java

```

package Admin;

import static Admin.Admin.createImagelcon;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Vector;
import javax.swing.ImageIcon;
import javax.swing.JButton;

```

```

import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableCellRenderer;

public class PartsReq extends ViewBooking{

    JPanel p3;
    JButton ref6,respondstatus;
    JLabel l_orderid,l_status;
    JTextField tf_orderid;
    JComboBox cb_status;
    String AID="1";int count1=0;
    Vector<Integer> productid=new Vector<>();
    PartsReq(){
        p3 = new JPanel();

        String icont1 = "/resources/partsreq 2.png";
        ImageIcon icon = createImageIcon(icont1);

        tp.addTab("Parts Required",icon,p3,"Manage OrderS");
        tp.setBackgroundAt(2, Color.WHITE);
        p3.setLayout(null);
        p3.setBackground(Color.WHITE);

        String icont2 = "/resources/refresh 2.jpg";
        ImageIcon b_icon = createImageIcon(icont2);

        ref6 = new JButton(b_icon);
        ref6.setText("REFRESH");
        ref6.setBounds(1100, 100, 120, 40);
        ref6.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                new ShowTable();
            }
        });
    }
}

```

```

p3.add(ref6);
ref6.setBackground(Color.WHITE);

l_orderid = new JLabel("OrderID ");
l_orderid.setBounds(1050,240,80,40);
p3.add(l_orderid);

tf_orderid = new JTextField();
tf_orderid.setBounds(1150,240,150,40);
p3.add(tf_orderid);

l_status = new JLabel("Status");
l_status.setBounds(1050,300,70,40);
p3.add(l_status);

String status[] = {"NONE","Packed","Shipped","Transit"};
cb_status = new JComboBox<String>(status);
cb_status.setBounds(1150,300,150,40);
p3.add(cb_status);

String icon3 = "/resources/updatestatus.png";
ImageIcon b_icon2 = createImageIcon(icon3);

respondstatus = new JButton(b_icon2);
respondstatus.setText("UPDATE STATUS");
respondstatus.setBounds(1100,350,190,40);
p3.add(respondstatus);
respondstatus.setBackground(Color.WHITE);
respondstatus.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try{
            ID = tf_orderid.getText();
            Object status = cb_status.getItemAt(cb_status.getSelectedIndex());
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update ecommerce set status = '"+status+"' where orderid
= '"+ID+"'";
            st.executeUpdate(query);

```

```

        con.close();
        new ShowTable();
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane, ex);
    }
}
});

try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

    Statement st = con.createStatement();
    String query = "select productid from items where adminid = '"+AID+"'";
    ResultSet rs=st.executeQuery(query);
    while(rs.next())
    {
        productid.add(rs.getInt(1));
    }
    Statement st1 = con.createStatement();
    for(int i=0;i<productid.size();i++)
    {
        String query1 = "select productid from ecommerce where productid =
 '"+productid.get(i)+"'";
        ResultSet rs1=st1.executeQuery(query1);
        while(rs1.next())
        {
            count1++;

        }
    }
    con.close();
}
catch(Exception ex){
    JOptionPane.showMessageDialog(rootPane, ex);
}
new ShowTable();
}

```

```

public class ShowTable {

    String [][]data;
    String[]colname;
    JTable jt1;
    ShowTable(){
        new Count();
        data = new String[count1][8];
        String [] heading =
{"OrderID","USERID","ProductID","Date","ProductName","Quantity","Totalprice","Status"};

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            int row = 0;
            System.out.println(productid);
            System.out.println(count1);
            for(int i=0;i<productid.size();i++)
            {
                String query = "select * from ecommerce where
productid='"+productid.get(i)+"'";
                ResultSet rs = st.executeQuery(query);
                while(rs.next())
                { System.out.println(productid);
                    data[row][0] = rs.getString("orderid");
                    data[row][1] = rs.getString("userid");
                    data[row][2] = rs.getString("productid");
                    java.util.Date date = Calendar.getInstance().getTime();
                    DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
                    String strDate = dateFormat.format(date);

                    data[row][3] = strDate;
                    data[row][4] = rs.getString("productname");
                    data[row][5] = rs.getString("quantity");
                    data[row][6] = rs.getString("price");
                    data[row][7] = rs.getString("status");
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

        row++;
    }
}
jt1 = new JTable(data,heading);
jt1.setBounds(10,30,1000,600);

jt1.setEnabled(false);

JScrollPane jsp = new JScrollPane(jt1);
jsp.setBounds(10,38,1014,612);

int height = jt1.getRowHeight();
jt1.setRowHeight(height+35);

//Aligning the text in JTable to center
DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
centerRenderer.setHorizontalAlignment( JLabel.CENTER );
for(int i=0;i<8;i++){
    jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
}
p3.add(jsp);
con.close();
}
catch(Exception e) {
    JOptionPane.showMessageDialog(p3,e.toString());
}
jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
jt1.setLayout(null);

}
}

public static void main(String args[]){
    new PartsReq();
}

}

```

Invoice.java

```
package Admin;

import static Admin.Admin.createImagelcon;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class Invoice extends PartsReq {
    JPanel p4;
    JLabel l1,l2,l3,l4,l5,l6,l7,l8,TotalLabel;
    JButton b1,b2;
    JTextField t1,t2,t3,t4,t5,t6;
    JTextArea ta;
    String
    partsPrice,serviceCharge,userId,vehicleNo,vehicleBrand,vehicleModel,serviceDone;
    Invoice(){
        p4 = new JPanel();

        String icont1 = "/resources/invoice.png";
        ImageIcon icon = createImagelcon(icont1);

        tp.addTab("INVOICE",icon,p4,"Generate Invoice");
        p4.setLayout(null);

        tp.setBackgroundAt(3, Color.WHITE);
        p4.setBackground(Color.WHITE);

        l1 = new JLabel("Appointment ID");
```



```
l1.setBounds(50,50,100,40);
l2 = new JLabel("Vehicle No");
l2.setBounds(50,100,100,40);
l3 = new JLabel("Vehicle Brand");
l3.setBounds(50,150,100,40);
l4 = new JLabel("Vehicle Model");
l4.setBounds(50,200,100,40);
l5 = new JLabel("Service Done");
l5.setBounds(50,250,100,40);
l6 = new JLabel("Price Of Parts");
l6.setBounds(50,400,100,40);
l7 = new JLabel("Service Charge");
l7.setBounds(50,450,100,40);
l8 = new JLabel("Total Amount");
l8.setBounds(50,500,100,40);
TotalLabel = new JLabel();
TotalLabel.setBounds(180,500,500,40);
```

```
String icont2 = "/resources/generatebill.png";
ImageIcon b_icon = createImageIcon(icont2);
```

```
b1 = new JButton(b_icon);
b1.setText("GENERATE BILL");
b1.setBounds(270,550,300,40);
b1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        partsPrice = t5.getText();
        serviceCharge = t6.getText();
        int PP = Integer.parseInt(partsPrice);
        int SC = Integer.parseInt(serviceCharge);
        int total = 0;
        total = PP + SC;
        String Total = String.valueOf(total);
        TotalLabel.setText(Total);

        try{
            //appointmentid = t1.getText();
            vehicleNo = t2.getText();
            vehicleBrand = t3.getText();
            vehicleModel = t4.getText();
            serviceDone = ta.getText();
```

```

        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st = con.createStatement();
        String query = "insert into invoice
(userid,VehicleNo,VehicleBrand,VehicleModel,ServiceDone,PartsPrice,ServiceCharge,
TotalAmount,adminid) values
('"+userid+"','"+vehicleNo+"','"+vehicleBrand+"','"+vehicleModel+"','"+serviceDone+"
','"+partsPrice+"','"+serviceCharge+"','"+Total+"','"+AID+"')";
        st.executeUpdate(query);
        String query2 = "update appointments set status = 'Delivered' where
appointmentid = '"+t1.getText()+"'";
        st.executeUpdate(query2);
        con.close();
        JOptionPane.showMessageDialog(rootPane,"Bill Generated
Successfully");
        t1.setText("");
        t2.setText("");
        t3.setText("");
        t4.setText("");
        t5.setText("");
        t6.setText("");
        ta.setText("");
        TotalLabel.setText("");

    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane,ex);
    }
}

});
b1.setBackground(Color.WHITE);

b2 = new JButton("Cancel");
b2.setBounds(60,550,90,40);
b2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        t1.setText("");
        t2.setText("");
    }
});

```

```
        t3.setText("");
        t4.setText("");
        t5.setText("");
        t6.setText("");
        ta.setText("");
        TotalLabel.setText("");
    }
});
b2.setBackground(Color.WHITE);

ta = new JTextArea();
ta.setBounds(180,250,500,130);
ta.setBorder(BorderFactory.createLineBorder(Color.GRAY,1));

t1 = new JTextField();
t1.setBounds(180,50,500,40);
t2 = new JTextField();
t2.setBounds(180,100,500,40);
t3 = new JTextField();
t3.setBounds(180,150,500,40);
t4 = new JTextField();
t4.setBounds(180,200,500,40);
t5 = new JTextField();
t5.setBounds(180,400,500,40);
t6 = new JTextField();
t6.setBounds(180,450,500,40);

p4.add(l1);
p4.add(l2);
p4.add(l3);
p4.add(l4);
p4.add(l5);
p4.add(l6);
p4.add(l7);
p4.add(l8);
p4.add(b1);
p4.add(b2);
p4.add(ta);
p4.add(t1);
```

```

        p4.add(t2);
        p4.add(t3);
        p4.add(t4);
        p4.add(t5);
        p4.add(t6);
        p4.add(TotalLabel);

    }

    public static void main(String args[]){
        new Invoice();
    }

}

```

ViewAllin.java

```

package Admin;

import static Admin.Admin.createImageIcon;
//import static
com.sun.xml.internal.fastinfoset.alphabet.BuiltInRestrictedAlphabets.table;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JButton;
//import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableCellRenderer;
public class ViewAllin extends Invoice{
    JPanel p5;
    JButton ref1;
}

```

```

ViewAllin(){

    p5 = new JPanel();

    String icont1 = "/resources/viewinvoice.jpg";
    ImageIcon icon = createImageIcon(icont1);

    tp.addTab("View All Invoice",icon,p5,"View All Invoices Generated");
    p5.setLayout(null);
    tp.setBackgroundAt(4, Color.WHITE);
    p5.setBackground(Color.WHITE);

    new ShowTable();

    String icont2 = "/resources/refresh 2.jpg";
    ImageIcon b_icon = createImageIcon(icont2);

    ref1 = new JButton(b_icon);
    ref1.setText("REFRESH");
    ref1.setBounds(1100, 100, 120, 40);
    ref1.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
            new ShowTable();

        }
    });

    p5.add(ref1);
    ref1.setBackground(Color.WHITE);

}

//class finding the number of records in a DataBase Table
public class Count{
    Count(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
            Statement st = con.createStatement();

```

```

        String query = "select count(*) from invoice where adminid='"+AID+"'";
        ResultSet rs = st.executeQuery(query);
        rs.next();
        count = rs.getInt(1);
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane,ex);
    }
}
}

```

```

public class ShowTable {

    String [][]data;
    String[] colname;
    JTable jt1;
    ShowTable(){
        new Count();
        data = new String[count][9];
        String [] heading =
{"SNo","USERID","VehicleNo","VehicleBrand","VehicleModel","ServiceDone","PartsP
rice","ServiceCharge","TotalAmount"};

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "select * from invoice where adminid='"+AID+"'";
            ResultSet rs = st.executeQuery(query);

            int row = 0;
            while(rs.next()) {
                data[row][0] = rs.getString("SNo");
                data[row][1] = rs.getString("USERID");
                data[row][2] = rs.getString("VehicleNo");
                data[row][3] = rs.getString("VehicleBrand");
                data[row][4] = rs.getString("VehicleModel");
                data[row][5] = rs.getString("ServiceDone");
                data[row][6] = rs.getString("PartsPrice");
            }
        }
    }
}

```

```

        data[row][7] = rs.getString("ServiceCharge");
        data[row][8] = rs.getString("TotalAmount");
        row++;
    }
    jt1 = new JTable(data,heading);
    jt1.setBounds(10,30,1000,600);
    jt1.setEnabled(false);
    JScrollPane jsp = new JScrollPane(jt1);
    jsp.setBounds(10,38,1014,612);

    int height = jt1.getRowHeight();
    jt1.setRowHeight(height+35);

//Aligning the text in JTable to center
    DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment( JLabel.CENTER );
    for(int i=0;i<9;i++){
        jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
    }

    p5.add(jsp);
    con.close();
}
catch(Exception e) {
    JOptionPane.showMessageDialog(p5,e.toString());
}
jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
jt1.setLayout(null);

}
}

public static void main(String args[]){
    new ViewAllin();
}
}

```

Custdetails.java

```
package Admin;

import static Admin.Admin.createImagelcon;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableCellRenderer;

public class Custdetail extends ViewAllin {
    JPanel p6;
    JButton ref4,blockuser;
    JLabel l_userid;
    JTextField tf_userid;
    String ID;

    Custdetail(){
        p6 = new JPanel();

        String icont1 = "/resources/managecust.jpg";
        ImageIcon icon = createImagelcon(icont1);

        tp.addTab("Manage Customer Details",icon,p6,"Block User");
        p6.setLayout(null);
        tp.setBackgroundAt(5, Color.WHITE);
        p6.setBackground(Color.WHITE);

        new ShowTable();
    }
}
```



```

String icont2 = "/resources/refresh 2.jpg";
ImageIcon b_icon = createImageIcon(icont2);

ref4 = new JButton(b_icon);
ref4.setText("REFRESH");
ref4.setBounds(1150, 100, 120, 40);
ref4.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        new ShowTable();
    }
});
p6.add(ref4);
ref4.setBackground(Color.WHITE);
//code
l_userid = new JLabel("UserID");
l_userid.setBounds(1110,200,50,40);
p6.add(l_userid);

tf_userid = new JTextField();
tf_userid.setBounds(1180,200,150,40);
p6.add(tf_userid);

blockuser = new JButton("BLOCK USER");
blockuser.setBounds(1140,250,150,40);
p6.add(blockuser);
blockuser.setBackground(Color.WHITE);
blockuser.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        try {
            ID = tf_userid.getText();
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update users set status = 'Blocked' where userid='"+ID+"'";
            st.executeUpdate(query);
            con.close();
            JOptionPane.showMessageDialog(rootPane,"User : "+ID+" Blocked ");
            new ShowTable();

```

```

    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane,ex);
    }
}
});
}

//class finding the number of records in a DataBase Table
public class Count{
    Count(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
            Statement st = con.createStatement();
            String query = "select count(*) from users";
            ResultSet rs = st.executeQuery(query);
            rs.next();
            count = rs.getInt(1);
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane,ex);
        }
    }
}

public class ShowTable {
    String [][]data;
    String[]colname;
    JTable jt1;
    ShowTable(){
        new Count();
        data = new String[count][10];
        String [] heading =
{"userid","username","password","mobilen","address1","address2","district","state
","pincode","status"};

        try {
            Class.forName("com.mysql.jdbc.Driver");

```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st = con.createStatement();
        String query = "select * from users";
        ResultSet rs = st.executeQuery(query);

        int row = 0;
        while(rs.next()) {
            data[row][0] = rs.getString("userid");
            //data[row][1] = rs.getString("name");
            data[row][1] = rs.getString("username");
            data[row][2] = rs.getString("password");
            data[row][3] = rs.getString("mobilenos");
            data[row][4] = rs.getString("address1");
            data[row][5] = rs.getString("address2");
            data[row][6] = rs.getString("district");
            data[row][7] = rs.getString("state");
            data[row][8] = rs.getString("pincode");
            data[row][9] = rs.getString("status");
            row++;
        }
        jt1 = new JTable(data,heading);
        jt1.setBounds(10,30,1000,600);

        jt1.setEnabled(false);

        JScrollPane jsp = new JScrollPane(jt1);
        jsp.setBounds(10,38,1064,612);

        int height = jt1.getRowHeight();
        jt1.setRowHeight(height+35);

        p6.add(jsp);
        con.close();
    }
    catch(Exception e) {
        JOptionPane.showMessageDialog(p6,e.toString());
    }
    jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
    jt1.setLayout(null);

```

```

//Aligning the text in JTable to center
    DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment( JLabel.CENTER );
    for(int i=0;i<10;i++){
        jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
    }

    }
}

    public static void main(String args[]){
        new Custdetail();
    }
}

```

DeliveryDetail.java

```

package Admin;

import static Admin.Admin.createImagelcon;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableCellRenderer;

public class DeliveryDetails extends Custdetail{

    JPanel p7;
    JButton ref5;

```

```
DeliveryDetails(){
```

```
    p7 = new JPanel();
```

```
    String icont1 = "/resources/delivery.png";
```

```
    ImageIcon icon = createImageIcon(icont1);
```

```
    tp.addTab("View Delivary Details",icon,p7,"Delivary Details");
```

```
    p7.setLayout(null);
```

```
    tp.setBackgroundAt(6, Color.WHITE);
```

```
    p7.setBackground(Color.WHITE);
```

```
    new ShowTable();
```

```
    String icont2 = "/resources/refresh 2.jpg";
```

```
    ImageIcon b_icon = createImageIcon(icont2);
```

```
    ref5 = new JButton(b_icon);
```

```
    ref5.setText("REFRESH");
```

```
    ref5.setBounds(1100, 100, 120, 40);
```

```
    ref5.addActionListener(new ActionListener(){
```

```
        public void actionPerformed(ActionEvent e){  
            new ShowTable();
```

```
        }
```

```
    });
```

```
    p7.add(ref5);
```

```
    ref5.setBackground(Color.WHITE);
```

```
}
```

```
//class finding the number of records in a DataBase Table
```

```
public class Count{
```

```
    Count(){
```

```
        try{
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            Connection con =
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram  
03");
```

```
            Statement st = con.createStatement();
```

```
            String query = "select count(*) from appointments where status =  
'Delivered'";
```

```

        ResultSet rs = st.executeQuery(query);
        rs.next();
        count = rs.getInt(1);
    }
    catch(Exception ex){
        JOptionPane.showMessageDialog(rootPane,ex);
    }
}
}

```

```

public class ShowTable {
String [][]data;
String[]colname;
JTable jt1;
ShowTable(){
    new Count();
    data = new String[count][8];
    String [] heading =
{"UserID","VehicleNo","AdminID","Service","Status","Timestamp","VehicleBrand","V
ehicleModel"};

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
        Statement st = con.createStatement();
        String query = "select * from appointments where status = 'Delivered'";
        ResultSet rs = st.executeQuery(query);

        int row = 0;
        while(rs.next()) {
            data[row][0] = rs.getString("userid");
            data[row][1] = rs.getString("vehicleno");
            data[row][2] = rs.getString("adminid");
            data[row][3] = rs.getString("service");
            data[row][4] = rs.getString("status");
            data[row][5] = rs.getString("timestramp");
            data[row][6] = rs.getString("vehiclebrand");
            data[row][7] = rs.getString("vehiclemodel");

```

```

        row++;
    }
    jt1 = new JTable(data,heading);
    jt1.setBounds(10,30,1000,600);

    jt1.setEnabled(false);

    JScrollPane jsp = new JScrollPane(jt1);
    jsp.setBounds(10,38,1014,612);

    int height = jt1.getRowHeight();
    jt1.setRowHeight(height+35);

//Aligning the text in JTable to center
    DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment( JLabel.CENTER );
    for(int i=0;i<8;i++){
        jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
    }

    p7.add(jsp);
    con.close();
}
catch(Exception e) {
    JOptionPane.showMessageDialog(p7,e.toString());
}
jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
jt1.setLayout(null);

}
}

public static void main(String args[]){
    new DeliveryDetails();
}
}

```

Profile.java

```
package Admin;

import static Admin.Admin.createImagelcon;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.Imagelcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Profile extends DeliveryDetails {
    JPanel p8;
    JLabel
adminid,adminname,mobilen,address1,address2,district,state,pincode,l_currentpas
s,l_newpass;
    JTextField
tf_adminid,tf_adminname,tf_mobileno,tf_address1,tf_address2,tf_district,tf_state,tf
_pincode,tf_currentpass,tf_newpass;
    JButton b_edit,b_submit,b_changepass,b_ref7,b_submit1;
    Profile(){
        p8 = new JPanel();

        String icont1 = "/resources/profile.jpg";
        Imagelcon icon = createImagelcon(icont1);

        tp.addTab("PROFILE",icon,p8,"View Your Profile");
        p8.setLayout(null);
        p8.setBackground(Color.WHITE);
        tp.setBackgroundAt(7,Color.WHITE);
    }
}
```



```
adminid = new JLabel("AdminID");
adminid.setBounds(100,50,100,40);
p8.add(adminid);
adminname = new JLabel("Name");
adminname.setBounds(100,120,100,40);
p8.add(adminname);
mobileno = new JLabel("MobileNo");
mobileno.setBounds(100,190,100,40);
p8.add(mobileno);
address1 = new JLabel("Address1");
address1.setBounds(100,260,100,40);
p8.add(address1);
address2 = new JLabel("Address2");
address2.setBounds(100,330,100,40);
p8.add(address2);
district = new JLabel("District");
district.setBounds(100,400,100,40);
p8.add(district);
state = new JLabel("State");
state.setBounds(100,470,100,40);
p8.add(state);
pincode = new JLabel("Pincode");
pincode.setBounds(100,540,100,40);
p8.add(pincode);

tf_adminid = new JTextField();
tf_adminid.setBounds(200,50,250,40);
p8.add(tf_adminid);
tf_adminid.setEditable(false);
tf_adminname = new JTextField();
tf_adminname.setBounds(200,120,250,40);
p8.add(tf_adminname);
tf_adminname.setEditable(false);
tf_mobileno = new JTextField();
tf_mobileno.setBounds(200,190,250,40);
p8.add(tf_mobileno);
tf_mobileno.setEditable(false);
tf_address1 = new JTextField();
tf_address1.setBounds(200,260,250,40);
tf_address1.setEditable(false);
p8.add(tf_address1);
```

```

tf_address2 = new JTextField();
tf_address2.setBounds(200,330,250,40);
p8.add(tf_address2);
tf_address2.setEditable(false);
tf_district = new JTextField();
tf_district.setBounds(200,400,250,40);
p8.add(tf_district);
tf_district.setEditable(false);
tf_state = new JTextField();
tf_state.setBounds(200,470,250,40);
p8.add(tf_state);
tf_state.setEditable(false);
tf_pincode = new JTextField();
tf_pincode.setBounds(200,540,250,40);
p8.add(tf_pincode);
tf_pincode.setEditable(false);

try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject", "root",
"Ram03");
    Statement st = con.createStatement();
    String query = "select * from admin where adminid = '"+AID+"'";
    ResultSet rs = st.executeQuery(query);
    if(rs.next()){
        tf_adminid.setText(rs.getString(1));
        tf_adminname.setText(rs.getString(2));
        tf_mobilenos.setText(rs.getString(5));
        tf_address1.setText(rs.getString(6));
        tf_address2.setText(rs.getString(7));
        tf_district.setText(rs.getString(8));
        tf_state.setText(rs.getString(9));
        tf_pincode.setText(rs.getString(10));
    }
    con.close();
}
catch(Exception ex){
    JOptionPane.showMessageDialog(rootPane,ex);
}

```

```
//Code for REfreshing the Entire ADmin Profile(Module)
b_ref7 = new JButton();
b_ref7.setText("REFRESH ALL");
b_ref7.setBounds(750,180,180,40);
p8.add(b_ref7);
b_ref7.setBackground(Color.WHITE);
b_ref7.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        dispose();
        new Profile();
    }
});
```

```
//Code for Edit Profile
b_submit = new JButton();
b_submit.setText("SUBMIT");
b_submit.setBounds(750,260,180,40);
p8.add(b_submit);
b_submit.setVisible(false);
b_submit.setBackground(Color.WHITE);
b_submit.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        b_edit.setVisible(true);
        b_submit.setVisible(false);
```

```
tf_adminname.setEditable(false);
tf_mobilenos.setEditable(false);
tf_address1.setEditable(false);
tf_address2.setEditable(false);
tf_district.setEditable(false);
tf_state.setEditable(false);
tf_pincode.setEditable(false);
```

```
String name,mobilenos,address1,address2,district,state,pincode;
name = tf_adminname.getText();
mobilenos = tf_mobilenos.getText();
address1 = tf_address1.getText();
//address2 = tf_address2.getText();
//district = tf_district.getText();
//state = tf_state.getText();
```

```

        //pincode = tf_pincode.getText();

        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "update admin set
name='"+name+"',mobilen='"+mobilen+"',address1='"+address1+" where
adminid='"+AID+"";
            st.executeUpdate(query);

//,address2='"+address2+"',district='"+district+"',state='"+state+"',pincode='"+pincode
+"

            con.close();
            JOptionPane.showMessageDialog(rootPane,"Changes Saved");
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane,ex);
        }

    }
});

b_edit = new JButton();
b_edit.setText("EDIT PROFILE");
b_edit.setBounds(750,260,180,40);
p8.add(b_edit);
b_edit.setVisible(true);
b_edit.setBackground(Color.WHITE);
b_edit.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        //tf_adminid.setEditable(true);
        tf_adminname.setEditable(true);
        tf_mobilen.setEditable(true);
        tf_address1.setEditable(false);
        //tf_address2.setEditable(true);
        //tf_district.setEditable(true);
        //tf_state.setEditable(true);
        //tf_pincode.setEditable(true);
    }
});

```

```

        b_edit.setVisible(false);
        b_submit.setVisible(true);
    }
});

//Code for Changing Password
b_changepass = new JButton();
b_changepass.setText("CHANGE PASSWORD");
b_changepass.setBounds(750,340,180,40);
p8.add(b_changepass);
b_changepass.setBackground(Color.WHITE);
b_changepass.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        l_currentpass.setVisible(true);
        l_newpass.setVisible(true);
        tf_currentpass.setVisible(true);
        tf_newpass.setVisible(true);
        b_submit1.setVisible(true);
    }
});

l_currentpass = new JLabel("Current Password");
l_currentpass.setBounds(600,420,150,40);
p8.add(l_currentpass);
l_currentpass.setVisible(false);

l_newpass = new JLabel("New Password");
l_newpass.setBounds(610,490,150,40);
p8.add(l_newpass);
l_newpass.setVisible(false);

tf_currentpass = new JTextField();
tf_currentpass.setBounds(750,420,200,40);
p8.add(tf_currentpass);
tf_currentpass.setVisible(false);

tf_newpass = new JTextField();
tf_newpass.setBounds(750,490,200,40);
p8.add(tf_newpass);
tf_newpass.setVisible(false);

```

```

b_submit1 = new JButton("SUBMIT");
b_submit1.setBounds(800,560,150,40);
p8.add(b_submit1);
b_submit1.setVisible(false);
b_submit1.setBackground(Color.WHITE);
b_submit1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        String currentpass,newpass,checkpass="";
        currentpass = tf_currentpass.getText();
        newpass = tf_newpass.getText();
        if(currentpass!=null && newpass!=null){
            try{
                Class.forName("com.mysql.jdbc.Driver");
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

                Statement st = con.createStatement();
                String query = "select * from admin where adminid='"+AID+"'";
                ResultSet rs = st.executeQuery(query);
                if(rs.next()){
                    checkpass = rs.getString("Passwords");
                }

                if(checkpass.equals(currentpass)){
                    String query1 = "update admin set passwords='"+newpass+"'where
adminid='"+AID+"'";
                    st.executeUpdate(query1);
                    JOptionPane.showMessageDialog(rootPane,"Password Changed
Successfully");
                    l_currentpass.setVisible(false);
                    l_newpass.setVisible(false);
                    tf_currentpass.setVisible(false);
                    tf_newpass.setVisible(false);
                    b_submit1.setVisible(false);

                }
                else{
                    JOptionPane.showMessageDialog(rootPane,"Password Mismatch");
                }
            }
            catch(Exception e){
                con.close();
            }
        }
    }
});

```

```

        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(rootPane,ex);
        }
    }

    }
    });
    JButton b_logout;

    b_logout = new JButton();
    b_logout.setText("LOG OUT");
    b_logout.setBounds(750,100,180,40);
    p8.add(b_logout);
    b_logout.setBackground(Color.WHITE);
    b_logout.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e){
            new Login();
        }
    });

}

public static void main(String args[]){
    new Profile();
}
}

```

Workers.java

```

package Admin;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.table.DefaultTableCellRenderer;

```

```

public class Workers extends Profile{
    JPanel p9;
    JButton b1,b2,b3,b4;
    JLabel l1,l2,l3,l4,l5,l6;
    JComboBox c1;JTextField t1,t2,t3;
    Workers()
    {
        p9=new JPanel();
        tp.addTab("Workers",p9);
        tp.setBackgroundAt(1,Color.WHITE);
        p9.setBackground(Color.WHITE);

        p9.setLayout(null);
        b1=new JButton("Refresh");
        b1.setBounds(1100, 100, 120, 40);
        b1.setBackground(Color.WHITE);

        b1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                new ShowTable();
            }
        });
        l1=new JLabel("WorkerID");
        l1.setBounds(1100,200,120,40);

        t1=new JTextField();
        t1.setBounds(1180,200,120,40);
        String[] str={"NONE","free","not free"};

        c1=new JComboBox(str);
        c1.setBounds(1050,250,120,40);
        c1.setBackground(Color.WHITE);

        l4=new JLabel("Add Workers");
        l4.setBounds(1050,330,120,40);

        l5=new JLabel("Worker Name");
        l5.setBounds(1050,380,120,40);

        t2=new JTextField();
    }
}

```



```

t2.setBounds(1150,380,120,40);

l6=new JLabel("Phone Number");
l6.setBounds(1050,430,120,40);

t3=new JTextField();
t3.setBounds(1150,430,120,40);

b4=new JButton("Add Workers");
b4.setBounds(1180,500,120,40);
b4.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

            Statement st = con.createStatement();
            String query = "insert into
workers(adminid,workername,workermobileno,workerstatus)
values('"+AID+"','"+t2.getText()+"','"+t3.getText()+"','free')";
            st.executeUpdate(query);
            try{
                Statement st1 = con.createStatement();
                String query1 = "select count(*) from workers where adminid = '"+AID+"'";
                ResultSet rs = st1.executeQuery(query1);
                rs.next();
                count = rs.getInt(1);
            }
            catch(Exception ex){
                System.out.println("e"+ex);
                //JOptionPane.showMessageDialog(rootPane,ex);
            }

            new ShowTable();
        }
        catch(Exception ex){
            System.out.println("ex "+ex);
            //JOptionPane.showMessageDialog(rootPane,ex);
        }
    }
}

```

```

    }

    }
});
b4.setBackground(Color.WHITE);

b2=new JButton("Change Status");
b2.setBounds(1180,250,120,40);
b2.addActionListener( new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
            Statement st = con.createStatement();
            String query = "update workers set
workerstatus='"+c1.getSelectedItem().toString()+"' where
workerid='"+t1.getText()+"'";
            st.executeUpdate(query);
            new ShowTable();
        }
        catch(Exception ex){
            System.out.println("e "+ex);
            //JOptionPane.showMessageDialog(rootPane,ex);
        }
    }
});
b2.setBackground(Color.WHITE);

p9.add(b1);p9.add(l1);p9.add(t1);p9.add(c1);p9.add(b2);p9.add(l4);p9.add(l5);p9.add
(l6);p9.add(b4);p9.add(t2);p9.add(t3);
p9.setBackground(Color.WHITE);
tp.setBackgroundAt(8,Color. WHITE);
try{
    Class.forName("com.mysql.jdbc.Driver");

```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

        Statement st = con.createStatement();
        String query = "select count(*) from workers where adminid = '"+AID+"'";
        ResultSet rs = st.executeQuery(query);
        rs.next();
        count = rs.getInt(1);
        System.out.println(count);
    }
    catch(Exception ex){
        System.out.println("e"+ex);
        //JOptionPane.showMessageDialog(rootPane,ex);
    }
    new ShowTable();
    setVisible(true);
}
public class ShowTable {
    String [][]data;
    String[]colname;
    JTable jt1;
    ShowTable(){

        data = new String[count][4];
        String [] heading = {"WorkerID","WorkerName","Mobile No","Status"};

        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
            Statement st = con.createStatement();
            String query = "select * from Workers where adminid= '"+AID+"'";
            ResultSet rs = st.executeQuery(query);

            int row = 0;
            while(rs.next()) {
                data[row][0] = rs.getString("workerid");

```

```

        data[row][1] = rs.getString("workername");
        data[row][2] = rs.getString("workermobileno");
        data[row][3] = rs.getString("workerstatus");
        row++;
    }
    jt1 = new JTable(data,heading);
    jt1.setBounds(100,100,1000,600);

    jt1.setEnabled(false);

    JScrollPane jsp = new JScrollPane(jt1);
    jsp.setBounds(10,38,1014,612);

    int height = jt1.getRowHeight();
    jt1.setRowHeight(height+35);

//Aligning the text in JTable to center
    DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment( JLabel.CENTER );
    for(int i=0;i<4;i++){
        jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);
    }

    p9.add(jsp);
    con.close();
}
catch(Exception e) {
    JOptionPane.showMessageDialog(p9,e.toString());
    System.out.println("sfa");
}
jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
jt1.setLayout(null);

}
}

public static void main(String args[]){
    new Workers();
}
}

```

Stock.java

```
package Admin;

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.table.DefaultTableCellRenderer;

public class Stock extends Workers{
    JPanel p10;
    JButton b1,b2,b3,b4;
    JLabel l1,l2,l3,l4,l5,l6,l7;
    JTextField c1;JTextField t1,t2,t3,t4;
    Stock()
    {
        p10=new JPanel();
        tp.addTab("Stock",p10);
        p10.setLayout(null);
        p9.setBackground(Color.WHITE);
        tp.setBackgroundAt(8,Color.WHITE);

        b1=new JButton("Refresh");
        b1.setBounds(1100, 100, 120, 40);
        b1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){

                new ShowTable();
            }
        });
        b1.setBackground(Color.WHITE);

        l1=new JLabel("ProductID");
        l1.setBounds(1100,200,120,40);

        t1=new JTextField();
```

```
t1.setBounds(1180,200,120,40);
String[] str={"NONE","free","not free"};

c1=new JTextField();
c1.setBounds(1050,250,120,40);
c1.setBackground(Color.WHITE);

b2=new JButton("Update Stock");
b2.setBounds(1180,250,120,40);
b2.setBackground(Color.WHITE);

l4=new JLabel("Add items");
l4.setBounds(1050,330,120,40);

l5=new JLabel("item Name");
l5.setBounds(1050,380,120,40);

t2=new JTextField();
t2.setBounds(1150,380,120,40);

l6=new JLabel("price");
l6.setBounds(1050,430,120,40);

t3=new JTextField();
t3.setBounds(1150,430,120,40);

l7=new JLabel("stock");
l7.setBounds(1050,480,120,40);

t4=new JTextField();
t4.setBounds(1150,480,120,40);

b4=new JButton("Add items");
b4.setBounds(1180,550,120,40);
b4.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");
```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
        Statement st = con.createStatement();
        String query = "insert into items(productname,price,stock,adminid)
values('"+t2.getText()+"','"+t3.getText()+"','"+t4.getText()+"','"+AID+"')";
        st.executeUpdate(query);
        try{
            Statement st1 = con.createStatement();
            String query1 = "select count(*) from items where adminid = '"+AID+"'";
            ResultSet rs = st1.executeQuery(query1);
            rs.next();
            count = rs.getInt(1);
        }
        catch(Exception ex){
            System.out.println("e"+ex);
            //JOptionPane.showMessageDialog(rootPane,ex);
        }

        new ShowTable();
    }
    catch(Exception ex){
        System.out.println("ex "+ex);
        //JOptionPane.showMessageDialog(rootPane,ex);
    }

}

});

b4.setBackground(Color.WHITE);

b2.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

```

```

        Statement st = con.createStatement();
        String query = "update items set stock='"+c1.getText()+"' where
productid='"+t1.getText()+"'";
        st.executeUpdate(query);
        new ShowTable();
    }
    catch(Exception ex){
        System.out.println("e"+ex);
        //JOptionPane.showMessageDialog(rootPane,ex);
    }
}
});
b2.setBackground(Color.WHITE);

p10.add(b1);p10.add(l1);p10.add(t1);p10.add(c1);p10.add(b2);p10.add(b4);p10.add(l
4);p10.add(l6);p10.add(l5);p10.add(t2);p10.add(t3);p10.add(t4);p10.add(l7);
p10.setBackground(Color.WHITE);
tp.setBackgroundAt(9,Color.WHITE);
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");

    Statement st = con.createStatement();
    String query = "select count(*) from items where adminid = '"+AID+"'";
    ResultSet rs = st.executeQuery(query);
    rs.next();
    count = rs.getInt(1);
    System.out.println(count);
}
catch(Exception ex){
    System.out.println("e"+ex);
    //JOptionPane.showMessageDialog(rootPane,ex);
}
new ShowTable();
setVisible(true);
}
public class ShowTable {
    String [][]data;

```



```

String[] colname;
JTable jt1;
ShowTable(){

    data = new String[count][4];
    String [] heading = {"ProductID","ProductName","Price","Stock"};

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/bcproject","root","Ram
03");
        Statement st = con.createStatement();
        String query = "select * from items where adminid= '"+AID+"'";
        ResultSet rs = st.executeQuery(query);

        int row = 0;
        while(rs.next()) {
            data[row][0] = rs.getString("productid");
            data[row][1] = rs.getString("productname");
            data[row][2] = rs.getString("price");
            data[row][3] = rs.getString("stock");
            row++;
        }
        jt1 = new JTable(data,heading);
        jt1.setBounds(100,100,1000,600);

        jt1.setEnabled(false);

        JScrollPane jsp = new JScrollPane(jt1);
        jsp.setBounds(10,38,1014,612);

        int height = jt1.getRowHeight();
        jt1.setRowHeight(height+35);

//Aligning the text in JTable to center
        DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
        centerRenderer.setHorizontalAlignment( JLabel.CENTER );
        for(int i=0;i<4;i++){
            jt1.getColumnModel().getColumn(i).setCellRenderer(centerRenderer);

```

```

        }

        p10.add(jsp);
        con.close();
    }
    catch(Exception e) {
        JOptionPane.showMessageDialog(p10,e.toString());
        System.out.println("s");
    }
    jt1.getColumnModel().getColumn(0).setPreferredWidth(50);
    jt1.setLayout(null);

}

}

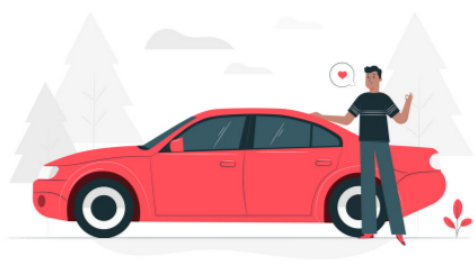
public static void main(String args[]){
    new Stock();
}

}

```


OUTPUT SCREENSHOTS

USER MODULE




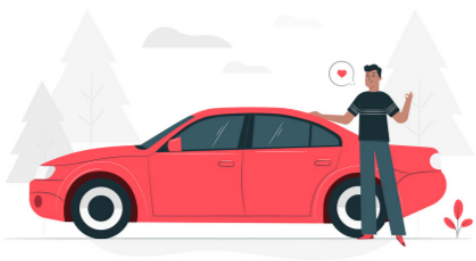
Welcome

LOGIN



REGISTER





Register

Full Name



AutoCare

User Name



autocare100

Password

.....

Confirm Password

.....

Mobile Number

7904601211

Address line 1

10, Gandhi Nagar

Address line 2

Kuniamuthur

State

Tamil Nadu

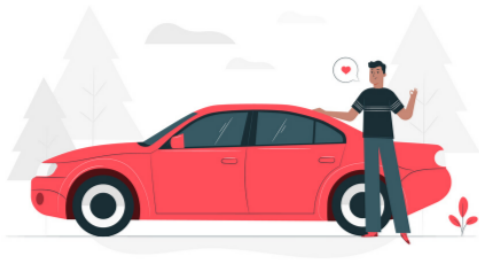
District

Coimbatore

Pincode

640125

REGISTER



LOGIN

USERNAME



autocare100

PASSWORD

.....

LOGIN >>

DASHBOARD >

ORDER ITEMS >

SCHEDULE >

CANCEL >

VIEW
ORDER >

LOGOUT



User Name autocare100

Mobile No 7904601211

Full Name AutoCare

Address
10, Gandhi Nagar
Kuniamuthur

District
Coimbatore

State
Tamil Nadu

Pincode
640125

YOUR PRESENT APPOINTMENTS

AP. ID	WORKSHOP	DATE	TIME

YOUR PRESENT RESPONSE

AS. ID	WORKSHOP	WORKER NAME	WORKER NO

VIEW

DASHBOARD >

ORDER ITEMS >

SCHEDULE >

CANCEL >

VIEW
ORDER >

LOGOUT

ORDER ITEMS



screw

PRIZE 400

AVAILABLE 114

QUANTITY
REQUIRED

5

ADD TO CART

CART

<input checked="" type="checkbox"/>	1	spanner	2	1600	3200
<input checked="" type="checkbox"/>	2	Audi tire	5	1600	8000
<input checked="" type="checkbox"/>	3	honda seat	1600	5	8000
<input checked="" type="checkbox"/>	4	engine oil	2	300	600
<input checked="" type="checkbox"/>	5	screw	5	400	2000
				Total:	21800

REMOVE

PLACE ORDER

DASHBOARD >

ORDER ITEMS >

SCHEDULE >

CANCEL >

VIEW
ORDER >

LOGOUT

IMMEDIATE RESPONSE

SERVICE



LOCATION

PINCODE

VEHICLE BRAND VEHICLE MODEL

RESPONSE

APPOINTMENTS

VEHICLE NO



VEHICLE BRAND



WORKSHOP

VEHICLE MODEL

 We will confirm pls wait 10 mins



WORKSHOP: Srirangam automobiles

WORKER NAME: raju

MOBILE NO: 9593941925

ok

APPOINTMENT

DASHBOARD >

ORDER ITEMS >

SCHEDULE >

CANCEL >

VIEW
ORDER >

LOGOUT

IMMEDIATE RESPONSE

SERVICE



LOCATION

PINCODE

VEHICLE BRAND VEHICLE MODEL

RESPONSE

APPOINTMENTS

VEHICLE NO



VEHICLE BRAND



WORKSHOP

VEHICLE MODEL



YOUR APPOINTMENT

BOOKED



ok

APPOINTMENT

DASHBOARD >

ORDER ITEMS >

SCHEDULE >

CANCEL >

VIEW
ORDER >

LOGOUT

IMMEDIATE RESPONSE

AS.ID



REASON

Auto-Care rules!

CANCEL

APPOINTMENTS

AP.ID



REASON

No, it does not

CANCEL

DASHBOARD>

ORDER ITEMS>

SCHEDULE >

CANCEL >

VIEW ORDER >

LOGOUT



User Name autocare100

Mobile No 7904601211

Full Name

AutoCare

Address

10, Gandhi Nagar
Kuniamuthur

District

Coimbatore

State

Tamil Nadu

Pincode

640125

YOUR PRESENT APPOINTMENTS

AP. ID	WORKSHOP	DATE	TIME
4	Srirangam automobiles	30/07/2020	2:00 pm

YOUR PRESENT RESPONSE

AS. ID	WORKSHOP	WORKER NAME	WORKER NO
5	Srirangam automobiles	raju	9593941925

VIEW

ASSIST >

APPOINTS >

As.ID	DATE AND TIME	WORKSHOP	STATUS
5	2020-08-02 17:26:46.853	Srirangam automobiles	cancelled

CLOSE

ASSIST >

APPOINTS >

As.ID	DATE AND TIME	WORKSHOP	STATUS
4	30/07/2020 2:00 pm	Srirangam automobiles	cancelled

CLOSE

DASHBOARD>

ORDER ITEMS>

SCHEDULE >

CANCEL >

VIEW
ORDER >

LOGOUT

YOUR ORDERS CANCELLATION

DELIVERY ID



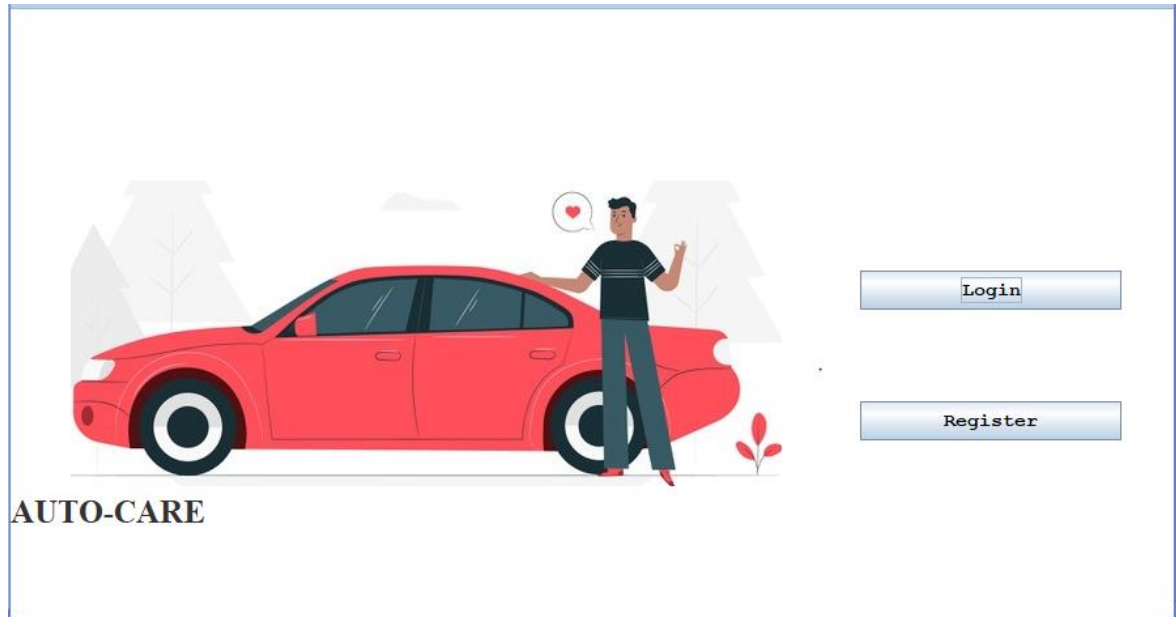
PRODUCT NAME



DELIVERY ID	no	Product	Quantity	Price
DI 62020-08-02 17:31:21.816	1	screw	4	400
DI 62020-08-02 17:31:21.816	2	spanner	4	1600
DI 62020-08-02 17:31:21.816	3	Audi tire	1600	3
DI 62020-08-02 17:31:21.816	4	Ferrari tire	3	500
DI 62020-08-02 17:31:21.816	5	engine oil	3	300

CANCEL

ADMIN MODULE



Admin Registration Form

Workshop Name	<input type="text" value="Srirangam Automobiles"/>
Workshop Username	<input type="text" value="ram prasath"/>
Mobile Number	<input type="text" value="7904601211"/>
Password	<input type="password" value="ramprasath19988"/>
Re-Enter Password	<input type="password" value="ramprasath1998"/>
Address line 1	<input type="text" value="65 mangamma nagar"/>
Address line 2	<input type="text" value="srirangam"/>
city	<input type="text" value="trichy"/>
state	<input type="text" value="Tamil Nadu"/>
pincode	<input type="text" value="620006"/>
<input checked="" type="checkbox"/> Accept Terms And Conditions.	
<input type="button" value="Register"/>	

Admin Name

ram prasath

Password

.....

Back

Login



Immediate Assistance

View Booking

Parts Required

INVOICE

View All Invoice

Manage Customer Details

View Delivery Details

PROFILE

Workers

Stock

Assist ID	User ID	Admin Id	Worker ID	landmark	Pincode	Vehicle Brand	Vehicle Model	Service	status	Estimated Time	Reason
1	1	1	15	srirangam	641025	Audi	A8L	Change glass	Approved	20	dapa
2	1	1	15	srirangam	641025	Audi	A8L	Full service	Declined		
3	1	1	21	ssa	641025	Audi	RS 7	Car wash	Declined	40	
4	5	1	1	srirangam	641025	Audi	A8L	Car wash	Approved		
5	6	1	1	srirangam	641025	Ferrari	Portofino	Full service	cancelled		Auto-Care rul...

REFRESH

AssistID1

AcceptDecline

Estimated Time to Reach The Location :

10In Minutes.

RESPOND USER

Immediate AssistanceView BookingParts RequiredINVOICEView All InvoiceManage Customer DetailsView Delivery DetailsPROFILEWorkersStock

UserID	VehicleNo	AppointmentID	Service	Status	Time And Date	VehicleBrand	VehicleModel
1	55776	1	Full service	Booked	4:00 pm 30/07/2020 ...	Audi	A8L
1	2334	2	Car wash	Approved	11:00 am 1/08/2020 ...	Ferrari	Portofino
5	66332	3	Change glass	Booked	10:00 am 1/08/2020 ...	Ferrari	Portofino
6	Tn3121	4	Full service	cancelled	2:00 pm 30/07/2020 ...	Audi	A8L

REFRESH

AppointmentID1

AcceptDecline

StatusOutForPickup

UPDATE STATUS

Immediate AssistanceView BookingParts RequiredINVOICEView All InvoiceManage Customer DetailsView Delivery DetailsPROFILEWorkersStock

OrderID	USERID	ProductID	Date	ProductName	Quantity	Totalprice	Status
DI 12020-08-01 ...	1	1	2020-08-02	screw	5	5	cancelled
DI 62020-08-02 ...	6	1	2020-08-02	screw	4	400	not delivered
DI 12020-08-01 ...	1	2	2020-08-02	screw driver	4	30	cancelled
DI 12020-08-01 ...	1	2	2020-08-02	screw driver	5	30	cancelled
DI 12020-08-01 ...	1	2	2020-08-02	screw driver	20	30	not delivered
DI 12020-08-01 ...	1	2	2020-08-02	screw driver	5	30	cancelled
DI 52020-08-02 ...	5	2	2020-08-02	honda tire A5	4	800	not delivered
DI 12020-08-01 ...	1	3	2020-08-02	hammer	100	5	cancelled
DI 52020-08-02 ...	5	3	2020-08-02	Audi tire	2	1600	cancelled
DI 62020-08-02 ...	6	3	2020-08-02	Audi tire	1600	3	not delivered
DI 12020-08-01 ...	1	4	2020-08-02	engine	9	1600	not delivered

REFRESH

OrderID

StatusNONE

UPDATE STATUS

Immediate Assistance

View Booking

Parts Required

INVOICE

View All Invoice

Manage Customer Details

View Delivery Details

PROFILE

Workers

Stock

userid	username	password	mobilen	address1	address2	district	state	pincode	status
1	rajeshkrish	krishnaraesh1998	7904601211	65	mangamma nag...	trichy	tamil nadu	620006	active
2	tamil2000	tamilxyz	9363211257	Devagar street	kuniamuttur	Coimbatore	Tamil Nadu	641041	active
3	rajesh100	Password1998	7904601211	65	mangamma nagar	coimbatore	Tamil Nadu	641025	Blocked
4	rajesh10000	Password1998	7904601211	65 mangamma n...	Srirangam	Kanyakumari	Tamil Nadu	641025	active
5	rajesh100000	password1998	7904601211	65 mangamma n...	srirangam	coimbatore	Karnataka	620006	active
6	autocare100	Password1998	7904601211	10, Gandhi Nagar	Kuniamuthur	Coimbatore	Tamil Nadu	640125	active

REFRESH

UserID

BLOCK USER

Immediate Assistance

View Booking

Parts Required

INVOICE

View All Invoice

Manage Customer Details

View Delivery Details

PROFILE

Workers

Stock

UserID	VehicleNo	AdminID	Service	Status	Timestamp	VehicleBrand	VehicleModel
--------	-----------	---------	---------	--------	-----------	--------------	--------------

REFRESH

Immediate Assistance

View Booking

Parts Required

INVOICE

View All Invoice

Manage Customer Details

View Delivery Details

PROFILE

Workers

Stock

AdminID

1

Name

Srirangam automobiles

MobileNo

9442422445

Address1

mangamma nagar

Address2

srirangam

District

Trichy

State

Tamil Nadu

Pincode

620006

LOG OUT

REFRESH ALL

EDIT PROFILE

CHANGE PASSWORD

Immediate Assistance

View Booking

Parts Required

INVOICE

View All Invoice

Manage Customer Details

View Delivery Details

PROFILE

Workers

Stock

ProductID	ProductName	Price	Stock
1	screw	400	110
2	honda tire A5	800	0
3	Audi tire	1600	123
4	engine	1600	100
5	audi seat	300	3
11	sdfa	32	23
12	sdfa	32	23

Refresh

ProductID

3

123

Update Stock

Add items

item Name

price

stock

Add items

Immediate Assistance

View Booking

Parts Required

INVOICE

View All Invoice

Manage Customer Details

View Delivery Details

PROFILE

Workers

WorkerID	WorkerName	Mobile No	Status
1	raju	9593941925	free
2	ram	9594941925	not free
3	shan	9598941925	not free
4	rakesh	9597941925	not free
5	kavin	9591941925	free
11	raju	9593941925	not free
12	ram	9594941925	not free
13	shan	9598941925	not free
14	rakesh	9597941925	not free
15	kavin	9591941925	not free
21	raju	9593941925	not free
22	ram	9594941925	not free

Refresh

WorkerID

NONE

Change Status

Add Workers

Worker Name

Phone Number

Add Workers

TEST CASES

USER TEST CASES

TEST CASE 1:

REGISTER

MODULE	TEST CASE	STATUS	REMARKS
Register	Name = Ashwin	Accepts and then enter user name.	Passed
Register	User name= AS2053S	Accepts and then enter the password.	Passed
Register	Password=ashwantha123\$	Accepts and then enter the confirm password.	Passed
Register	Confirm Password=ashwantha123\$	Accepts and then enter the mobile number.	Passed
Register	Mobile Number =9597941925	Accepts and then enter the Address Line 1	Passed
Register	Address Line 1 = Krishna Street	Accepts and then enter the Address Line 2	Passed
Register	Address Line 2 = Devanga Nagar	Accepts and then ask to select the state	Passed
Register	State = Tamil Nadu	Accepts and then ask to select the district	Passed
Register	District = Coimbatore	Accepts and then enter the pincode	Passed
Register	Pincode = 641025	Accepts and then redirects to Home Screen Page	Passed

TEST CASE 2:**REGISTER**

MODULE	TEST CASE	STATUS	REMARKS
Register	Name = Rajesh	Accepts and then enter user name.	Passed
Register	User name= Rajesh2000\$\$5	Accepts and then enter the password.	Passed
Register	Password=rajesh\$\$dev	Accepts and then enter the confirm password.	Passed
Register	Confirm Password=rajesh\$\$dev	Accepts and then enter the mobile number.	Passed
Register	Mobile Number =9597941925	Accepts and then enter the Address Line 1	Passed
Register	Address Line 1 = Kuniyamuttur	Accepts and then enter the Address Line 2	Passed
Register	Address Line 2 = Edayarpalayam	Accepts and then ask to select the state	Passed
Register	State = Tamil Nadu	Accepts and then ask to select the district	Passed
Register	District = Coimbatore	Accepts and then enter the pincode	Passed
Register	Pincode = 641041	Accepts and then redirects to Home Screen Page	Passed

TEST CASE 3:**REGISTER**

MODULE	TEST CASE	STATUS	REMARKS
Register	Name = Rajesh\$123	Accepts and Display "Name contains only alphabets"	Failed
Register	User name= Rajesh2000\$\$5	Accepts and then enter the password.	Passed
Register	Password=rajesh\$\$dev	Accepts and then enter the confirm password.	Passed
Register	Confirm Password=rajesh\$\$dev	Accepts and then enter the mobile number.	Passed
Register	Mobile Number =9597941925	Accepts and then enter the Address Line 1	Passed
Register	Address Line 1 = Kuniyamuttur	Accepts and then enter the Address Line 2	Passed
Register	Address Line 2 = Edayarpalayam	Accepts and then ask to select the state	Passed
Register	State = Tamil Nadu	Accepts and then ask to select the district	Passed
Register	District = Coimbatore	Accepts and then enter the pincode	Passed
Register	Pincode = 641041	Accepts and then redirects to Home Screen Page	Passed

TEST CASE 4:**REGISTER**

MODULE	TEST CASE	STATUS	REMARKS
Register	Name = Kavın	Accepts and then enter user name.	Passed
Register	User name= Kavin2000\$\$5	Accepts and then enter the password.	Passed
Register	Password=kavin\$\$dev	Accepts and then enter the confirm password.	Passed
Register	Confirm Password=kavin1995	Accepts and then display "Password and Confirm Password are mismatched"	Failed
Register	Mobile Number =9597941925	Accepts and then enter the Address Line 1	Passed
Register	Address Line 1 = Kuniyamuttur	Accepts and then enter the Address Line 2	Passed
Register	Address Line 2 = Edayarpalayam	Accepts and then ask to select the state	Passed
Register	State = Tamil Nadu	Accepts and then ask to select the district	Passed
Register	District = Coimbatore	Accepts and then enter the pincode	Passed
Register	Pincode = 641041	Accepts and then redirects to Home Screen Page	Passed

TEST CASE 5:**REGISTER**

MODULE	TEST CASE	STATUS	REMARKS
Register	Name = Kavın	Accepts and then enter user name.	Passed
Register	User name= Kavın2000\$\$5	Accepts and then enter the password.	Passed
Register	Password=kavın\$\$dev	Accepts and then enter the confirm password.	Passed
Register	Confirm Password=kavın\$\$dev	Accepts and then enter the mobile no	Passed
Register	Mobile Number =955556	Accepts and then display "Mobile no can be of only 10 digits"	Failed
Register	Address Line 1 = Kuniyamuttur	Accepts and then enter the Address Line 2	Passed
Register	Address Line 2 = Edayarpalayam	Accepts and then ask to select the state	Passed
Register	State = Tamil Nadu	Accepts and then ask to select the district	Passed
Register	District = Coimbatore	Accepts and then enter the pincode	Passed
Register	Pin code = 641041	Accepts and then redirects to Home Screen Page	Passed

TEST CASE 6:**REGISTER**

MODULE	TEST CASE	STATUS	REMARKS
Register	Name = Kavin	Accepts and then enter user name.	Passed
Register	User name= Kavin2000\$\$5	Accepts and then enter the password.	Passed
Register	Password=kavin\$\$dev	Accepts and then enter the confirm password.	Passed
Register	Confirm Password=kavin\$\$dev	Accepts and then enter the mobile no	Passed
Register	Mobile Number =9555562235	Accepts and then enter the Address Line 1	Passed
Register	Address Line 1 = Kuniyamuttur	Accepts and then enter the Address Line 2	Passed
Register	Address Line 2 = Edayarpalayam	Accepts and then ask to select the state	Passed
Register	State = Tamil Nadu	Accepts and then ask to select the district	Passed
Register	District = Coimbatore	Accepts and then enter the pincode	Passed
Register	Pincode = 64104	Accepts and then displays “pincode is of only 6 digits”	Failed

TEST CASE 7 :**LOGIN**

MODULE	TEST CASE	STATUS	REMARKS
Login	User name=Kavin2000\$\$5	Accepts and then enter the password	Passed
Login	Password=kavin\$\$dev	Accepts and then go to the Dashboard	Passed

TEST CASE 8 :**LOGIN**

MODULE	TEST CASE	STATUS	REMARKS
Login	User name=AS2053S	Accepts and then enter the password	Passed
Login	Password=ashwantha123\$	Accepts and then go to the Dashboard	Passed

TEST CASE 9 :**LOGIN**

MODULE	TEST CASE	STATUS	REMARKS
Login	User name=AS2053S	Accepts and then enter the password	Passed
Login	Password=ashwantha123	Accepts and display "Incorrect username or password"	Failed

TEST CASE 10 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=1	Accepts and then select the service needed	Passed
Schedule	Service=Repair	Accepts and then enter the location	Passed
Schedule	Location=12,Pallani Nagar, Edayarpalayam, Coimbatore, Tamil Nadu	Accepts and then enter the pincode	Passed
Schedule	pincode=641006	Accepts and then select the vehicle brand	Passed
Schedule	Vehicle brand=Audi	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A3	Accepts and assist booked	Passed

TEST CASE 11 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=2	Accepts and then select the service needed	Passed
Schedule	Service=Engine Failure	Accepts and then enter the location	Passed
Schedule	Location=13,DevangaNagar, Edayarpalayam, Coimbatore, Tamil Nadu	Accepts and then enter the pincode	Passed
Schedule	pincode=641025	Accepts and then select the vehide brand	Passed
Schedule	Vehicle brand=Honda	Accepts and then select the vehide model	Passed
Schedule	Vehicle model = A1	Accepts and assist booked	Passed

TEST CASE 12 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=1	Accepts and display "Already response registered"	Failed
Schedule	Service=Repair	Accepts and then enter the location	Passed
Schedule	Location=12,Pallani Nagar, Edayarpalayam, Coimbatore, Tamil Nadu	Accepts and then enter the pincode	Passed
Schedule	pincode=641006	Accepts and then select the vehicle brand	Passed
Schedule	Vehicle brand=Audi	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A3	Accepts and assist booked	Passed

TEST CASE 13 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=3	Accepts and then select the service	passed
Schedule	Service=Repair	Accepts and then enter the location	Passed
Schedule	Location=12,Kuniyamuttur Coimbatore, Tamil Nadu	Accepts and then enter the pincode	Passed
Schedule	pincode=6410067	Accepts and then displays “pincode can be of 6 digits only”	Failed
Schedule	Vehicle brand=Audi	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A3	Accepts and assist booked	Passed

TEST CASE 14 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=1	Accepts and then select the service needed	Passed
Schedule	Service=Repair	Accepts and then select the date	Passed
Schedule	Date=21/06/2020	Accepts and then select the time	Passed
Schedule	Time=10:00	Accepts and then select the vehicle brand	Passed
Schedule	Vehicle brand=Audi	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A3	Accepts and appointment booked	Passed

TEST CASE 15 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=2	Accepts and then select the service needed	Passed
Schedule	Service=Engine Failure	Accepts and then select the date	Passed
Schedule	Date=22/06/2020	Accepts and then select the time	Passed
Schedule	Time=11:00	Accepts and then select the vehicle brand	Passed
Schedule	Vehicle brand=Honda	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A1	Accepts and appointment booked	Passed

TEST CASE 16 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=1	Accepts and then select the service	Passed
Schedule	Service=Repair	Accepts and then select the date	Passed
Schedule	Date=21/06/2020	Accepts and then display "Already you have booked an appointment for this date"	Failed
Schedule	Time=10:00	Accepts and then select the vehicle brand	Passed
Schedule	Vehicle brand=Audi	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A3	Accepts and appointment booked	Passed

TEST CASE 17 :

MODULE	TEST CASE	STATUS	REMARKS
Schedule	User id=1	Accepts and then select the service	Passed
Schedule	Service=Repair	Accepts and then select the date	Passed
Schedule	Date=22/06/2020	Accepts and then select the time	Passed
Schedule	Time=11:00	Accepts and then display "Already this time slot is booked by another customer"	Failed
Schedule	Vehicle brand=Audi	Accepts and then select the vehicle model	Passed
Schedule	Vehicle model = A3	Accepts and appointment booked	Passed

TEST CASE 18 :**CANCEL ASSIST**

MODULE	TEST CASE	STATUS	REMARKS
cancel	User id=1	Accepts and then select the assist id	Passed
cancel	Assist id=2	Accepts and then enter the reason	Passed
cancel	Reason="The problem was rectified by myself.."	Accepts and the assist id gets deleted	Passed

TEST CASE 19 :**CANCEL ASSIST**

MODULE	TEST CASE	STATUS	REMARKS
cancel	User id=1	Accepts and then select the assist id	Passed
cancel	Assist id=2	Accepts and then enter the reason	Passed
cancel	Reason=" "	Accepts and displays "Enter the reason pls"	Failed

TEST CASE 20 :**CANCEL ASSIST**

MODULE	TEST CASE	STATUS	REMARKS
cancel	User id=1	Accepts and then select the Appointment id	Passed
cancel	Appointment id=2	Accepts and then enter the reason	Passed
cancel	Reason="I cant come to the workshop due to heavy rain."	Accepts and the appointment id gets deleted	Passed

TEST CASE 21 :**CANCEL ASSIST**

MODULE	TEST CASE	STATUS	REMARKS
cancel	User id=1	Accepts and then select the Appointment Id	Passed
cancel	Appointment id=2	Accepts and then enter the reason	Passed
cancel	Reason=" "	Accepts and displays "Enter the reason pls"	Failed

TEST CASE 22 :**ORDER ITEMS**

MODULE	TEST CASE	STATUS	REMARKS
Order items	User id=1	Accepts and then select the item	Passed
Order items	Item = screw	Accepts and then enter the quantity	Passed
Order items	Quantity = 5	Accepts and gets placed in the cart	Passed
Order items	Item = hammer	Accepts and then enter the quantity	Passed
Order items	Quantity = 3	Accepts and gets placed in the cart	Passed
Order items	Item = tire Honda A5	Accepts and then enter the quantity	Passed
Order items	Quantity = 4	Accepts and gets placed in the cart	Passed

Order items	Item = Honda seat	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed
Order items	Item = Engine	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed

TEST CASE : 23

ORDER ITEMS

MODULE	TEST CASE	STATUS	REMARKS
Order items	User id=5	Accepts and then select the item	Passed
Order items	Item = tire Honda A5	Accepts and then enter the quantity	Passed
Order items	Quantity = 4	Accepts and gets placed in the cart	Passed
Order items	Item = Honda seat	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed
Order items	Item = Engine	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed

TEST CASE 24 :

ORDER ITEMS

MODULE	TEST CASE	STATUS	REMARKS
Order items	User id=2	Accepts and then select the item	Passed
Order items	Item = screw	Accepts and then enter the quantity	Passed
Order items	Quantity = 5	Accepts and gets placed in the cart	Passed
Order items	Item = hammer	Accepts and then enter the quantity	Passed
Order items	Quantity = 0	Accepts and displays "Cant add to Cart"	Failed

Order items	Item = tire Honda A5	Accepts and then enter the quantity	Passed
Order items	Quantity = 4	Accepts and gets placed in the cart	Passed
Order items	Item = Honda seat	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed
Order items	Item = Engine	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed

TEST CASE 25 :

ORDER ITEMS

MODULE	TEST CASE	STATUS	REMARKS
Order items	User id=5	Accepts and then select the item	Passed
Order items	Item = screw	Accepts and then enter the quantity	Passed
Order items	Quantity = 5	Accepts and gets placed in the cart	Passed
Order items	Item = hammer	Accepts and then enter the quantity	Passed
Order items	Quantity = 3	Accepts and gets placed in the cart	Passed
Order items	Item = tire Honda A5	Accepts and then enter the quantity	Passed
Order items	Quantity = 4	Accepts and gets placed in the cart	Passed
Order items	Item = Honda seat	Accepts and then enter the quantity	Passed
Order items	Quantity = 1	Accepts and gets placed in the cart	Passed
Order items	Item = Engine	Accepts and then enter the quantity	Passed

Order items	Quantity = 1	Accepts and gets placed in the cart	Passed
Order items	Item = Oil	Accepts and displays “Cart is full, 5 items can be placed in an order”	Failed
Order items	Quantity = 1	Accepts and displays “Cart is full, 5 items can be placed in an order”	Failed

TEST CASE 26 :

CANCEL ORDER

MODULE	TEST CASE	STATUS	REMARKS
Cancel order	User id=5	Accepts and then select the order id	Passed
Cancel order	Order id =A5 12/06/20 15:05:03	Accepts and then select the product	Passed
Cancel order	Product =screw	Accepts and cancels the order placed	Passed

TEST CASE 27:

CANCEL ORDER

MODULE	TEST CASE	STATUS	REMARKS
Cancel order	User id=1	Accepts and then select the order id	Passed
Cancel order	Order id =A5 21/06/20 03:05:03	Accepts and then select the product	Passed
Cancel order	Product =hero tire	Accepts and cancels the order placed	Passed

TEST CASE 28 :**CANCEL ORDER**

MODULE	TEST CASE	STATUS	REMARKS
Cancel order	User id=1	Accepts and then select the order id	Passed
Cancel order	Order id ='	Accepts and then displays "select the order id"	Failed
Cancel order	Product ='"	Accepts and then displays "select the order id"	Failed

TEST CASE 29:**CANCEL ORDER**

MODULE	TEST CASE	STATUS	REMARKS
Cancel order	User id=5	Accepts and then select the order id	Passed
Cancel order	Order id =A5 21/06/20 03:05:03	Accepts and then select the product	Passed
Cancel order	Product ='"	Accepts and then displays "select the product"	Failed

ADMIN TEST CASES

TEST CASE –1:

MODULE	TESTCASE	STATUS	REMARKS
REGISTRATION	Name=Nandha	Accepts and asks admin to enter username	Passed
REGISTRATION	UserName = Nandha2000	Accepts and asks admin to enter Mobilenno	Passed
REGISTRATION	Mobilenno = 9786835368	Accpets and asks admin to enter Password	Passed
REGISTRATION	Password = Password1998	Accpets and asks admin to REenter Password	Passed
REGISTRATION	REenter Password = Password1998	Accepts and asks admin to enter Address Line1	Passed
REGISTRATION	Address Line 1 = 127,RameyGounder layout	Accepts and asks admin to enter Address Line 2	Passed
REGISTRATION	Address Line 2 = Thennampalayam	Accepts and asks admin to enter city	Passed
REGISTRATION	City = Tirupur	Accepts and asks admin to enter State	Passed
REGISTRATION	State = TamilNadu	Accepts and asks admin to enter Pincode	Passed
REGISTRATION	Pincode = 641604	Accepts and asks admin to enable Terms and condition checkbox	Passed
REGISTRATION	Accept terms and condition checkbox to checked-TRUE	Redirects to MainPage	Passed

TEST CASE –2:

MODULE	TEST CASE	STATUS	REMARKS
REGISTRATION	Name = Nandha	Accepts and asks admin to enter user name	Passed
REGISTRATION	Username = Nandha 2000	Accepts and asks admin to enter Mobilenno	Passed
REGISTRATION	Mobile No = 968827	Displays Error Message saying that the entered mobile no is not valid	Passed
REGISTRATION	Password = Password1998	Accepts and asks admin to REenterPassword	Passed
REGISTRATION	REenter Password = Password1998	Accepts and asks admin to enter Address Line1	Passed
REGISTRATION	Address Line 1 = 127,RameyGounder layout	Accepts and asks admin to enter Address Line 2	Passed
REGISTRATION	Address Line 2 = Thennampalayam	Accepts and asks admin to enter city	Passed
REGISTRATION	City = Tirupur	Accepts and asks admin to enter State	Passed
REGISTRATION	State = TamilNadu	Accepts and asks admin to enter Pincode	Passed
REGISTRATION	Pincode = 641604	Accepts and asks admin to enable Terms and condition checkbox	Passed
REGISTRATION	Accept terms and condition checkbox to checked-TRUE	Redirects to MainPage	Passed

TEST CASE –3:

MODULE	TEST CASE	STATUS	REMARKS
REGISTRATION	Name = Nandha	Accepts and asks admin to enter user name	Passed
REGISTRATION	Username = Nandha 2000	Accepts and asks admin to enter Mobileno	Passed
REGISTRATION	Mobile No = 9688279307	Accepts and asks for Password	Passed
REGISTRATION	Password = Password1998	Accepts and asks admin to REenterPassword	Passed
REGISTRATION	REenter Password = Password1997	Displays an error message “Password Mismatch”	Failed
REGISTRATION	Address Line 1 = 127,RameyGounder layout	Accepts and asks admin to enter Address Line 2	Passed
REGISTRATION	Address Line 2 = Thennampalayam	Accepts and asks admin to enter city	Passed
REGISTRATION	City = Tirupur	Accepts and asks admin to enter State	Passed
REGISTRATION	State = TamilNadu	Accepts and asks admin to enter Pincode	Passed
REGISTRATION	Pincode = 641604	Accepts and asks admin to enable Terms and condition checkbox to MainPage	Passed
REGISTRATION	Accept terms and condition checkbox to checked-TRUE	Redirects to MainPage	Passed

TEST CASE –4:

MODULE	TEST CASE	STATUS	REMARKS
REGISTRATION	Name = Nandha	Accepts and asks admin to enter user name	Passed
REGISTRATION	Username = Nandha 2000	Accepts and asks admin to enter Mobilen0	Passed
REGISTRATION	Mobile No = 9688279307	Accepts and asks for Password	Passed
REGISTRATION	Password = Pass	Displays an error message "Password must be 8 characters long"	Failed
REGISTRATION	REnter Password = Pass	Displays an Error Message	Failed
REGISTRATION	Address Line 1 = 127,RameyGounder layout	Accepts and asks admin to enter Address Line 2	Passed
REGISTRATION	Address Line 2 = Thennampalayam	Accepts and asks admin to enter city	Passed
REGISTRATION	City = Tirupur	Accepts and asks admin to enter State	Passed
REGISTRATION	State = TamilNadu	Accepts and asks admin to enter Pincode	Passed
REGISTRATION	Pincode = 641604	Accepts and asks admin to enable Terms and condition checkbox	Passed
REGISTRATION	Accept terms and condition checkbox to checked-TRUE	Redirects to MainPage	Passed

TEST CASE –5:

MODULE	TEST CASE	STATUS	REMARKS
REGISTRATION	Name = Nandha	Accepts and asks admin to enter user name	Passed
REGISTRATION	Username = Nandha 2000	Accepts and asks admin to enter Mobilenno	Passed
REGISTRATION	Mobile No = 9688279307	Accepts and asks for Password	Passed
REGISTRATION	Password = Password1998	Accepts and asks admin to REenterPassword	Passed
REGISTRATION	REenter Password = Password1998	Accepts and asks Admin to enter Address Line 1	Passed
REGISTRATION	Address Line 1 = 127,RameyGounder layout	Accepts and asks admin to enter Address Line 2	Passed
REGISTRATION	Address Line 2 = Thennampalayam	Accepts and asks admin to enter city	Passed
REGISTRATION	City = Tirupur	Accepts and asks admin to enter State	Passed
REGISTRATION	State = TamilNadu	Accepts and asks admin to enter Pincode	Passed
REGISTRATION	Pincode = 6416	Displays an error message “INVALID PINCODE”	Failed
REGISTRATION	Accept terms and condition checkbox to checked-TRUE	Redirects to MainPage	Passed

TEST CASE –6:

MODULE	TESTCASE	STATUS	REMARKS
REGISTRATION	Name=Nandha	Accepts and asks admin to enter username	Passed
REGISTRATION	UserName = Nandha2000	Accepts and asks admin to enter Mobilen0	Passed
REGISTRATION	Mobilen0 = 9786835368	Accpets and asks admin to enter Password	Passed
REGISTRATION	Password = Password1998	Accpets and asks admin to REenter Password	Passed
REGISTRATION	REenter Password = Password1998	Accepts and asks admin to enter Address Line1	Passed
REGISTRATION	Address Line 1 = 127,RameyGounder layout	Accepts and asks admin to enter Address Line 2	Passed
REGISTRATION	Address Line 2 = Thennampalayam	Accepts and asks admin to enter city	Passed
REGISTRATION	City = Tirupur	Accepts and asks admin to enter State	Passed
REGISTRATION	State = TamilNadu	Accepts and asks admin to enter Pincode	Passed
REGISTRATION	Pincode = 641604	Accepts and asks admin to enable Terms and condition checkbox	Passed
REGISTRATION	Accept terms and condition checkbox to to checked-FALSE	Displays error Message and asks admin to enable Terms and condition checkbox	Failed

TEST CASE –7:

MODULE	TEST CASE	STATUS	REMARKS
LOGIN	Username = Nandha2000	Accepts and asks admin to enter password	Passed
LOGIN	Password = Password1998	Redirects to admin profile	Passed

TEST CASE –8:

MODULE	TEST CASE	STATUS	REMARKS
LOGIN	Username = Rajesh200	Accepts and asks admin to enter password	Passed
LOGIN	Password = AUtocare200	Redirects to admin profile	Passed

TEST CASE –9:

MODULE	TEST CASE	STATUS	REMARKS
LOGIN	Username = Nandha2000	Accepts and asks admin to enter password	Passed
LOGIN	Password = Password1997	Displays an error message showing “Please enter valid user name and passowrd”	Failed

TEST CASE –10:

MODULE	TEST CASE	STATUS	REMARKS
IMMEDIATE ASSISTANCE TAB	AssistID = 1	Accepts and asks admin to acknowledge	Passed
IMMEDIATE ASSISTANCE TAB	Accepts	Status Updated and estimated time combo box is shown	Passed
IMMEDIATE ASSISTANCE TAB	Declines	Status Updated	Passed

TEST CASE –11:

MODULE	TEST CASE	STATUS	REMARKS
IMMEDIATE ASSISTANCE TAB	AssistID = 5(Which is Invalid)	Accepts and asks admin to acknowledge	Passed
IMMEDIATE ASSISTANCE TAB	Accepts	Displays an error "AssisID not found"	Failed
IMMEDIATE ASSISTANCE TAB	Declines	Displays an error "AssisID not found"	Failed

TEST CASE –12:

MODULE	TEST CASE	STATUS	REMARKS
VIEW BOOKING TAB	AppointmentID = 1	Accepts and asks admin to acknowledge	Passed
VIEW BOOKING TAB	Accepts	Status Updated	Passed
VIEW BOOKING TAB	Declines	Status Updated	Passed

TEST CASE –13:

MODULE	TEST CASE	STATUS	REMARKS
VIEW BOOKING TAB	AppointmentID = 5(Which is Invalid)	Accepts and asks admin to acknowledge	Passed
VIEW BOOKING TAB	Accepts	Displays an error message "AppointmentID not found"	Passed
VIEW BOOKING TAB	Declines	Displays an error message "AppointmentID not found"	Passed

TEST CASE –14:

MODULE	TEST CASE	STATUS	REMARKS
PARTS REQ TAB	OrderID = 1	Accepts and asks admin to update status	Passed
PARTS REQ TAB	Status = Packed (From ComboBox)	Status Updated	Passed

TEST CASE –15:

MODULE	TEST CASE	STATUS	REMARKS
PARTS REQ TAB	OrderID = 5(Which is Invalid)	Displays an error message "OrderID not found"	Failed
PARTS REQ TAB	Status = Packed (From ComboBox)	Displays an error message "OrderID not found"	Failed

TEST CASE –16:

MODULE	TEST CASE	STATUS	REMARKS
CUST DETAIL TAB	UserID = 1	Accepts and asks admin to block the user	passed

TEST CASE-17:

MODULE	TEST CASE	STATUS	REMARKS
CUST DETAIL TAB	UserID = 5(which is invalid)	Displays an error message “UserID not found”	Failed

TEST CASE –18:

MODULE	TEST CASE	STATUS	REMARKS
WORKERS TAB	WorkerID = 1	Accepts and asks admin to update status	Passed
WORKERS TAB	Status = “Free”(From comboBox)	Status Updated	Passed

TEST CASE –19:

MODULE	TEST CASE	STATUS	REMARKS
WORKERS TAB	WorkerID = 6(which is invalid)	Displays an error message “WorkerID not found”	Failed
WORKERS TAB	Status = “Free”(From comboBox)	Displays an error message “WorkerID not found”	Failed

TEST CASE –20:

MODULE	TEST CASE	STATUS	REMARKS
STOCK TAB	ProductID = 1	Accepts and ask admin to update stock	Passed
STOCK TAB	Stock = 25	Stock Updated	Passed

TEST CASE –21:

MODULE	TEST CASE	STATUS	REMARKS
STOCK TAB	ProductID = 21(which is invalid)	Displays an error message “ProductID not found”	Failed
STOCK TAB	Stock = 25	Displays an error message “ProductID not found”	Failed

TARGET

There has always been a whole lot of difficulties surrounding the AutoMobile industry when it comes to servicing them regularly to ensure that the vehicles are indeed working like a charm.

Automobile repair centers in India, in our opinion are not as efficient as they should be and could be if they utilized the current gen Technologies.

We are using those said technologies to create a web of Workshops and common people to make it as easy as Amazon did with delivering products. And we also provide what we call as door step assists when users get into a metaphorical mudhole.